Localized LQG Optimal Control for Large-Scale Systems

Yuh-Shyang Wang and Nikolai Matni

***Abstract*— This paper presents a scalable linear quadratic Gaussian (LQG) synthesis algorithm for large-scale localizable systems. These are systems for which the effect of each local process noise and sensor noise can be localized in closed loop despite communication delays between controllers. In particular, the resulting localized LQG (LLQG) controller can be synthesized and implemented in a localized way using local plant model information, which is extremely favorable for large-scale interconnected systems. This is achieved by combining the alternating direction method of multipliers (ADMM) algorithm with localized linear quadratic regulator (LLQR) decomposition for state feedback. Simulations show that for certain systems, the LLQG controller, with its constraints on locality, settling time, and delays, can achieve transient performance similar to an idealized centralized $\mathcal{H}_2$ optimal controller. The algorithm is demonstrated on a randomized heterogeneous system with about 10^4 states, where the distributed and centralized methods cannot be computed within a reasonable amount of time.**

I. Introduction

Large-scale networked systems permeate both modern life and academic research, with familiar examples including the Internet, smart grid, wireless sensor networks, and biological networks in science and medicine. The scale of these systems pose fundamental challenge for controller design: simple controllers designed completely locally cannot guarantee global stability, whereas a traditional centralized controller is neither scalable to compute, nor physically implementable. The field of distributed (decentralized) optimal control has emerged to address the realistic communication constraints amongst local sensors, actuators, and controllers into the design process for these networked systems.

A distributed optimal control problem is commonly formulated by imposing an information sharing constraint on the stabilizing controller. The tractability of the problem depends on the relations between the information sharing constraint and the plant, and it can be NP-hard in the worst case for certain structures [1], [2]. It has been shown that the distributed optimal control problem admits a convex reformulation in the Youla domain if [3] and only if [4] the information sharing constraint is quadratically invariant (QI) with respect to the plant. With the identification of quadratic invariance as an appropriate means of convexifying the distributed optimal control problem, computationally tractable solutions for various types of distributed constraints and objectives are provided in [5]–[11].

Although the distributed optimal control problem with a QI constraint is computationally tractable, both the synthesis and the implementation of the QI controller are *not scalable* for large systems. In fact, for a plant with a strongly connected topology, even as simple as a chain, the QI condition implies that local measurements need to be shared among *all local controllers* in the network (which can be inferred from [12]). This then implies that finding an optimal sparse controller K (the transfer function from measurement to control action) for a strongly connected networked system cannot be formulated in a convex manner in the Youla domain. Regularization [13], [14], convex relaxation [15], and spatial truncation [16] have been used in hopes of finding a distributed controller that is scalable to implement. However, not only are these methods sub-optimal, but the synthesis procedure is still centralized. In particular, these methods involve solving a large-scale optimization problem with the knowledge of global plant model beforehand, which is not scalable.

In our previous works [17]–[19], we introduced the notion of localizable system in which a localized closed loop response exists. For a localizable system, the controller achieving the desired closed loop response can always be implemented in a localized way if we allow the communication of both *measurements* and *controller states* between local controllers. In addition to localized implementation, we further show that linear quadratic regulator (LQR) problems for state feedback localizable systems can be *solved* in a localized way [18]. Specifically, we show that the localized LQR (LLQR) problem can be decomposed into several local optimization problems, in which each local optimization problem can be solved in an independent and parallel way using only local plant model information. However, the technique does not extend to output feedback localizable systems [19] directly due to the coupling constraint between feedback and estimation.

In this paper, we combine the technique of distributed optimization and LLQR decomposition to tackle the coupling constraint for output feedback localizable systems. The resulting localized linear quadratic Gaussian (LLQG) controller also has favorable properties for large-scale systems, including localized implementation and localized synthesis using local plant model information.

The rest of this paper is structured as follows. Section II introduces the system model, the traditional distributed optimal control formulation, and our prior works on lo-

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

calizable systems. In Section III, we introduce the LLQR decomposition for a state feedback problem. The result is extended to output feedback in Section IV, where we utilize the technique of alternating direction method of multipliers (ADMM) [20] to locally synthesize the LLQG controller. To demonstrate the effectiveness of our method, we synthesize the LLQG controller for a randomized heterogeneous system with about 10^4 states in Section V. The conclusions and future research direction are summarized in Section VI.

II. Preliminaries and Prior Works

This section starts with the interconnected system model we considered in this paper. We then explain why the controller design problem for such system is difficult using the traditional distributed optimal control formulation, and introduce some necessary background on our prior works.

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 sub-systems, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ encodes the interaction between these sub-systems. Each sub-system i is associated with a state vector x_i, control action u_i, and measurement y_i. We assume that the dynamics of each sub-system i is given in the form of

$$\begin{aligned} x_i[k+1] &= A_{ii}x_i[k] + B_{ii}u_i[k] + \sum_{j \in \mathcal{N}_i} A_{ij}x_j[k] + \delta x_i[k] \\ y_i[k] &= C_{ii}x_i[k] + \delta y_i[k], \end{aligned} \quad (1)$$

where $A_{ii}, A_{ij}, B_{ii}, C_{ii}$ are constant matrices with compatible dimensions, δx_i and δy_i the perturbation on state and measurement, respectively, and $\mathcal{N}_i$ the (upstream) neighboring set of i defined by $\mathcal{N}_i = \{j | (j, i) \in \mathcal{E}\}$. The following is a simple example in the form of (1), which is used to illustrate several ideas throughout this paper.

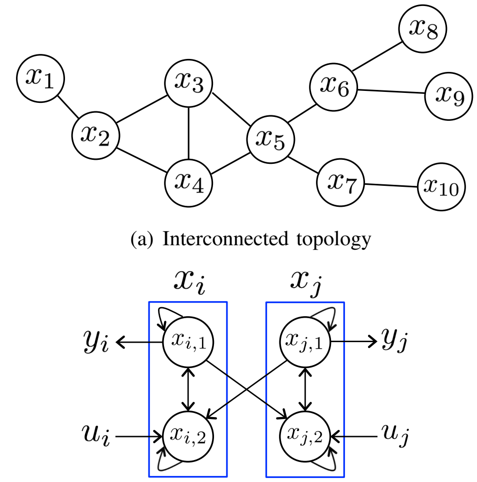
Example 1: Consider a network system with interconnected topology given by Fig. 1(a). Each sub-system x_i is assumed to have two interacting scalar states $x_{i,1}, x_{i,2}$, a sensor on $x_{i,1}$, and an actuator on $x_{i,2}$. The interaction between two neighboring sub-systems is illustrated in Fig. 1(b), where $x_{i,1}$ can affect $x_{j,2}$ if i and j are neighbor. This model is inspired from the second-order dynamics in power network or mechanical system. Specifically, consider a second-order equation

$$m_i \ddot{\theta}_i + d_i \dot{\theta}_i = -\sum_{j \in \mathcal{N}_i} k_{ij}(\theta_i - \theta_j) + w_i + u_i$$

for some scalar θ_i. If we define the state vector by $x_i = [\theta_i \quad \dot{\theta}_i]^\top$, take the measurement on θ_i, and use $e^{A\Delta t} \approx I + A\Delta t$ to discretize the plant, we get a discretized plant model with interaction graph described by Fig. 1(b).

B. Challenges to Traditional Method

The interconnected systems pose great challenges to controller design when (i) the effect of sub-system interaction A_{ij} cannot be neglected, and (ii) the number of sub-systems n grows large. The distributed method and the completely local method can handle one of the situations above, but not both.



(a) Interconnected topology

(b) Interaction between neighboring sub-systems

Fig. 1. Illustrative Example

By completely local method, for an extreme case, the controller is implemented *and* designed in a purely local way. Specifically, each local control rule is given by $u_i = K_i y_i$, where K_i is designed with the local plant model (A_{ii}, B_{ii}, C_{ii}) *only*. Clearly, such control scheme is scalable to arbitrary large n, but does not guarantee global stability if the effect of A_{ij} is not negligible.

To show that the distributed optimal control method is not scalable to large n, we construct the global plant model as

$$\begin{aligned} x[k+1] &= Ax[k] + B_2u[k] + \delta_x[k] \\ y[k] &= C_2x[k] + \delta_y[k] \end{aligned} \quad (2)$$

where x, u, y, δ_x, and δ_y are stacked vectors of local state, control, measurement, process disturbance, and sensor disturbance, respectively. The global plant model (A, B_2, C_2) can be derived by combining the matrices in (1) in an appropriate way. Equation (2) can be written in state space form as

$$P = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & \mathbf{0} & D_{12} \\ C_2 & D_{21} & \mathbf{0} \end{array}\right] = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$$

for some matrices C_1, D_{12}, B_1, D_{21} representing the cost function and noise dynamics, and $P_{ij} = C_i(zI - A)^{-1}B_j + D_{ij}$. Assume a dynamic output feedback control law $u = Ky$. The distributed optimal control problem is commonly formulated as

$$\begin{aligned} \underset{K}{\text{minimize}} \quad & ||P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}|| \\ \text{subject to} \quad & K \text{ stabilizes } P \\ & K \in \mathcal{S}_c \end{aligned} \quad (3)$$

where $\mathcal{S}_c$ is a subspace constraint that K must satisfy in order to be implementable on the communication network of the system.

In order to have a scalable controller implementation, it is natural to impose sparsity constraints on K. For example, one may want the control action u_1 at node 1 in Fig. 1(a) to use the measurements y_i from node $i = 1, \dots, 5$ only. This is equivalently saying that the delay from measurements y_i for node $i = 6, \dots, 10$ to the control action u_1 at node 1 is set to be infinity. This information sharing constraint is not QI, and (3) becomes non-convex. In fact, using the delay interpretation of QI in [12], the sparsity constraint for a plant with strongly connected topology is not QI in general. Although the non-convex problem corresponding to a sparse $\mathcal{S}_c$ may be solved via convex relaxation [15] or other techniques, (3) (or the modification of (3)) is still a large-scale optimization problem depending on a global plant model, which cannot be solved in a scalable way.

C. Localized Distributed Control

In our previous works, we point out that the distributed controller does not need to be implemented directly by K, so K need not be sparse. Instead, we introduce four closed loop transfer matrices (R, M, N, L) to link the relation between closed loop response, controller implementation, and controller synthesis. For a system described by (2), the closed loop transfer matrices (R, M, N, L) are defined by

$$\begin{bmatrix} x \\ u \end{bmatrix} = \begin{bmatrix} R & N \\ M & L \end{bmatrix} \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix}. \tag{4}$$

The *localized* distributed optimal control problem is then formulated by

$$\underset{\{R,M,N,L\}}{\text{minimize}} \quad || \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} R & N \\ M & L \end{bmatrix} \begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} || \tag{5a}$$

$$\text{subject to} \quad \begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} R & N \\ M & L \end{bmatrix} = \begin{bmatrix} I & 0 \end{bmatrix} \tag{5b}$$

$$\begin{bmatrix} R & N \\ M & L \end{bmatrix} \begin{bmatrix} zI - A \\ -C_2 \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix} \tag{5c}$$

$$\begin{bmatrix} zR & zN \\ zM & L \end{bmatrix} \in \mathcal{S} \cap \mathcal{RH}_\infty \tag{5d}$$

where z is the variable of z-transform, $\mathcal{RH}_\infty$ the set of real rational stable proper transfer matrices, and $\mathcal{S}$ a spatio-temporal constraint on the closed loop transfer matrices. The following theorem is proposed in [19] and extended in [21].

Theorem 1: If (5b) - (5d) is feasible, then the desired closed loop response (4) can be achieved by the controller implementation

$$\begin{aligned} z\beta &= \tilde{R}^+ \beta + \tilde{N} y \\ u &= \tilde{M} \beta + L y \end{aligned} \tag{6}$$

where $\tilde{R}^+ = z(I - zR)$, $\tilde{N} = -zN$, $\tilde{M} = zM$, L are stable proper transfer matrices, β the controller state. The implementation (6) is internally stabilizing.

The implementation (6) is similar to the state space realization of the controller, but we generalize the state space matrices into stable proper transfer matrices $(\tilde{R}^+, \tilde{N}, \tilde{M}, L)$. If (5b) - (5d) is feasible for a sparse $\mathcal{S}$, Theorem 1 suggests that the controller achieving the desired closed loop response can also be implemented in a localized way by (6).

Example 2: For Fig. 1(a), if there exists a control law such that the state x_1 and control action u_1 at node 1 is affected only by disturbances δx_i and δy_i from node $i = 1, \dots, 5$ in closed loop. Then, using Theorem 1, the control action u_1 at node 1 can be computed by collecting measurements *and* controller states (y_i, β_i) from node $i = 1, \dots, 5$ to achieve the desired closed loop response.

We emphasize the fact that both the synthesis and implementation of (5a) - (5d) are done in the *closed loop transfer matrices domain*, which avoids the non-convexity in (3). This method can also be considered as a generalization of disturbance localization [22] and the decoupled control law [23]. Specifically, a centralized controller is required for the method in [22], and the method in [23] only works when the interconnected systems have a non-overlapping partition (block diagonal sparsity constraint on R). On the other hand, Theorem 1 can handle *overlapping* localized regions, which arise when there are communication delays between local controllers.

However, the optimization problem (5a) - (5d) does not decompose naturally due to the coupling between constraints (5b) and (5c). The goal of this paper is to provide a scalable algorithm to *solve* (5a) - (5d) in a localized way with (5a) being the LQG cost. In this paper, the spatio-temporal constraint $\mathcal{S}$ in (5d) is given before synthesis. Some simple rules to *design* $\mathcal{S}$ and the actuator locations (B_2) are proposed in [24].

III. LLQR Decomposition

In this section, we consider the LLQR decomposition technique for a state feedback localizable system. This decomposition technique plays an important role on the decomposition of the output feedback problem. The content in this section is a generalization of LLQR control proposed in [18]. Specifically, the spatio-temporal constraint discussed in this section is not limited to (d, T) localized finite impulse response (FIR) constraint as in [18]. The formulation and the cost function are also more general.

A. Column-wise Decomposition

Consider a state feedback plant model with $C_2 = I$, $D_{21} = \mathbf{0}$, and $B_1 = I$. The assumption $B_1 = I$ is made due to the simplicity of presentation, while the method described in this section works as long as the process noise correlates locally, i.e. B_1 is block diagonal. Consider the following problem

$$\begin{aligned} \underset{\{R,M\}}{\text{minimize}} \quad & || \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} R \\ M \end{bmatrix} ||_{\mathcal{H}_2}^2 \\ \text{subject to} \quad & \begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} R \\ M \end{bmatrix} = I \\ & \begin{bmatrix} R \\ M \end{bmatrix} \in \mathcal{S} \cap \frac{1}{z} \mathcal{RH}_\infty. \end{aligned} \tag{7}$$

This is the general formulation of the LLQR problem in [18]. Note that (7) admits a *column-wise* decomposition.

Specifically, the objective function in (7) can be rewritten as

$$\sum_{j=1}^{n} || \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} R \\ M \end{bmatrix}_j ||_{\mathcal{H}_2}^2$$

where the subscript j represents the j-th block column of $[R^\top M^\top]^\top$ that corresponds to the closed loop response from δx_j to (x, u). The constraints in (7) decompose column-wise as well. Denote $\mathcal{S}_j$ the convex set constraint on block column j. We can solve

$$\begin{aligned} \underset{\{R_j, M_j\}}{\text{minimize}} \quad & || \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} R \\ M \end{bmatrix}_j ||_{\mathcal{H}_2}^2 \\ \text{subject to} \quad & \begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} R \\ M \end{bmatrix}_j = [I]_j \\ & \begin{bmatrix} R \\ M \end{bmatrix}_j \in \mathcal{S}_j \cap \frac{1}{z}\mathcal{RH}_\infty. \end{aligned} \tag{8}$$

for $j = 1, \ldots, n$ instead of solving (7). The benefit of this decomposition is that we can possibly reduce the dimension of (8) *from global scale to local scale* when $\mathcal{S}_j$ is highly sparse. Due to the decomposition property of the $\mathcal{H}_2$ norm, we can analyze the closed loop response for each local disturbance δx_j in its own localized region in a parallel and independent way, even when their localized regions are *overlapping*. Note that (8) is a block column-wise decomposition for each subsystem. We can also perform a column-wise decomposition for each *single scalar state* if necessary.

B. Localized Region

We now focus on a specific j in (8). Recall that $\mathcal{S}_j$ is a constraint on the closed loop response from δx_j to x and u. Therefore, the block row-wise sparsity pattern of $\mathcal{S}_j$ encodes the set of affected states and activated control actions under disturbance δx_j. We can construct a localized region, local state vector, and local plant model based on this information. Without introducing complicated notation and definition, we try to explain the concept in an intuitive way. Consider the following illustrative example.

Example 3: Consider Fig. 1(a) with state feedback (full sensing). Assume that the disturbance δx_1 at node 1 is only allowed to affect the state x_1 and x_2 at node 1 and node 2 in closed loop. In addition, we want to achieve this using the control action u_i from node $i = 1, \ldots, 4$. In this case, a localized region can be constructed by Region(x_1) = $\{x_1, x_2, x_3, x_4\}$ as shown in Fig. 2. The boundary of this localized region is the set $\{x_3, x_4\}$. Intuitively, by keeping the boundary being zero for all time, there is no way for the disturbance to *escape* from the localized region through state propagation. In addition, the effects of all control actions are limited within the localized region. We can then solve (8) using the local state, control, and plant model defined within this localized region.

One simple rule to construct a localized region is taking the union of the following three sets.

(i) The affected states ($\{x_1, x_2\}$ in Example 3)
(ii) The states that can be affected by the activated control actions directly ($\{x_1, \ldots, x_4\}$ in Example 3)

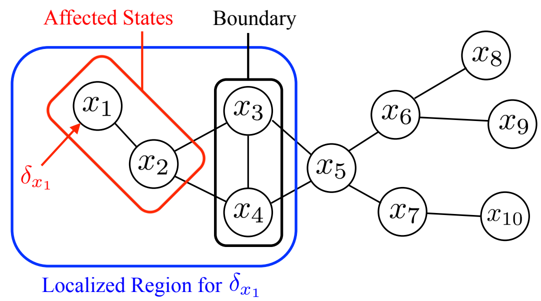


Fig. 2. Localized Region

(iii) The states that are connected from the affected states, i.e. the set $\{x_i | A_{ik} \neq 0, x_k \in$ affected states$\}$. ($\{x_1, \ldots, x_4\}$ in Example 3)

For a localized region, the external region Ext(.) is defined as its complement set. The boundary of a localized region is defined as the set of states that connect the localized region toward the external region, i.e. $\{x_k | A_{ik} \neq 0, x_k \in$ Region$(x_j), x_i \in$ Ext$(x_j)\}$. Intuitively, the affected states can be considered as the *internal region* of the localized region, and the boundary is the set that *separates* the internal region from the external region. As long as the boundary states are keeping zero for all time, and the control actions do not affect the external region (which is ensured by construction rule (ii)), the solution of (8) can be analyzed completely within the localized region. Construction rule (iii) ensures that the boundary states are included in the localized region.

With the notion of localized region, we define the local state vector x_ℓ the stacked vector of x_i for all $x_i \in$ Region(x_j). The local control vector u_ℓ is stacked vector of all activated local control action u_i. The local plant model $(A_\ell, B_{2\ell})$ associated with Region(x_j) is defined by selecting submatrices of (A, B_2) consisting of the columns and rows associated with x_ℓ and u_ℓ. Let e_ℓ be the initial condition of the source of disturbance. The local convex set constraint $\mathcal{S}_\ell$ can be defined by selecting the block rows of $\mathcal{S}_j$ associated with $[x_\ell^\top u_\ell^\top]^\top$, and column corresponding to e_ℓ. With this dimension reduction, each LLQR sub-problem (8) can be expressed in the reduced dimension with the form

$$\begin{aligned} \underset{\{x_\ell, u_\ell\}}{\text{minimize}} \quad & || \begin{bmatrix} C_{1\ell} & D_{12\ell} \end{bmatrix} \begin{bmatrix} x_\ell \\ u_\ell \end{bmatrix} ||_{\mathcal{H}_2}^2 \\ \text{subject to} \quad & \begin{bmatrix} zI - A_\ell & -B_{2\ell} \end{bmatrix} \begin{bmatrix} x_\ell \\ u_\ell \end{bmatrix} = e_\ell \\ & \begin{bmatrix} x_\ell \\ u_\ell \end{bmatrix} \in \mathcal{S}_\ell \cap \frac{1}{z}\mathcal{RH}_\infty. \end{aligned} \tag{9}$$

Example 4: Consider the example in Fig. 2. We can solve (9) with local state, control, and plant model defined associated with its localized region. Recall in Fig. 1(b) that x_1 has two single scalar state $x_1 = [x_{1,1}^\top x_{1,2}^\top]^\top$. The initial condition needs to be set on each single scalar state in δx_1. Specifically, we set $x_{1,1}[1] = 1$ first and compute the solution (x_ℓ^1, u_ℓ^1), with the convex set constraint $\mathcal{S}_\ell^1$ corresponding to the initial condition. We then set $x_{1,2}[1] = 1$ and compute the solution x_ℓ^2, u_ℓ^2. The solution of (8) for $j = 1$, which has two columns, is then given by

$$\begin{bmatrix} x_\ell^1 \\ u_\ell^1 \end{bmatrix} \text{ and } \begin{bmatrix} x_\ell^2 \\ u_\ell^2 \end{bmatrix}$$

for each column in the reduced dimension.

Note that each solution of (9) only depends local plant model. Furthermore, if we impose FIR constraint on x_ℓ and u_ℓ, (9) becomes a finite dimensional convex program. This is due to the fact that the $\mathcal{H}_2$ norm of a FIR transfer matrix G can be computed by its spectral components as

$$||G||_{\mathcal{H}_2}^2 = \sum_{i=0}^{T} \text{Trace}(G[i]^\top G[i]) \tag{10}$$

where T is the length of FIR. If we have communication delays constraint, we impose different sparsity constraint on different spectral components of (x_ℓ, u_ℓ) in (9).

IV. Localized LQG Synthesis

In this section, we solve the output feedback problem of (5a) - (5d) with a LQG cost. This is done by combining the technique of ADMM with LLQR decomposition.

A. ADMM Algorithm

We first assume that the process noise and sensor noise only correlate locally. For simplicity of presentation, let

$$\begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} = \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & \sigma_y I \end{bmatrix} \tag{11}$$

where σ_y is the relative magnitude between process disturbance and sensor disturbance. The method described in this section also works when (11) is block diagonal. The corresponding LLQG problem is given by

$$\begin{aligned} \underset{\{R,M,N,L\}}{\text{minimize}} \quad & || \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} R & \sigma_y N \\ M & \sigma_y L \end{bmatrix} ||_{\mathcal{H}_2}^2 & (12) \\ \text{subject to} \quad & (5b) - (5d). & (13) \end{aligned}$$

We assume that the constraint (5d) is a sparse FIR constraint. This ensures that the optimization problem is finite dimensional and amenable to LLQR decomposition. Without constraint (5c), the optimization problem admits a column-wise LLQR decomposition, which can be solved in a localized way. On the other hand, we can also verify the feasibility of (5c) in a localized way through a row-wise LLQR decomposition. It is then natural to use ADMM algorithm to iteratively force the solutions of these two different problems converging to the global optimal solution.

In the following, we use

$$CL = \begin{bmatrix} R & N \\ M & L \end{bmatrix}$$

to represent the closed loop transfer matrix we want to optimize for. We define extended-real-value functions $f(CL_1)$ and $g(CL_2)$ by

$$f(CL_1) = \begin{cases} (12) & \text{if } (5b), (5d) \\ \infty & \text{otherwise} \end{cases}$$

$$g(CL_2) = \begin{cases} 0 & \text{if } (5c), (5d) \\ \infty & \text{otherwise} \end{cases}$$

Problem (13) can then be equivalently formulated as

$$\begin{aligned} \underset{\{CL_1, CL_2\}}{\text{minimize}} \quad & f(CL_1) + g(CL_2) \\ \text{subject to} \quad & CL_1 = CL_2. \end{aligned} \tag{14}$$

Problem (14) can be solved via the standard ADMM approach [20] as

$$CL_1^{k+1} = \underset{CL_1}{\text{argmin}} \left(f(CL_1) + \frac{\rho}{2} ||CL_1 - CL_2^k + \Lambda^k||_{\mathcal{H}_2}^2 \right) \tag{15a}$$

$$CL_2^{k+1} = \underset{CL_2}{\text{argmin}} \left(g(CL_2) + \frac{\rho}{2} ||CL_2 - CL_1^{k+1} - \Lambda^k||_{\mathcal{H}_2}^2 \right) \tag{15b}$$

$$\Lambda^{k+1} = \Lambda^k + CL_1^{k+1} - CL_2^{k+1} \tag{15c}$$

where the square of $\mathcal{H}_2$ norm is a shorthand of (10). Subproblem (15a) can be solved in a localized way using column-wise LLQR decomposition on the objective and the constraint. Similarly, subproblem (15b) can also be solved in a localized way using row-wise LLQR decomposition. Equation (15c) can be decomposed either column-wise or row-wise. The only variable that needs to be shared globally before synthesis is ρ in (15a) - (15b). Other than this variable, (15a) - (15c) is a localized algorithm for LLQG synthesis using column-wise and row-wise decomposition alternatively and iteratively.

In the following, we derive the conditions to ensure the convergence of CL_1^k and CL_2^k in (15a) - (15c). We then show that (15a) - (15b) can be solved in closed form. Specifically, the solution of (15a) - (15b) can be expressed as an affine function of its reference solution. Therefore, we only need to solve (15a) - (15b) in closed form *once*, and then update (15a) - (15c) as a linear dynamical system. This suggests that the LLQG synthesis problem can be solved almost as fast as the LLQR problem.

It should be noted that the ADMM method can handle other objective function, as long as it admits a column-wise decomposition, or a row-wise decomposition, or a combination of the two. An interesting case is that we can solve for arbitrary B_1 and D_{21} when $[C_1 \quad D_{12}]$ is block diagonal. This means that the LLQG controller can be synthesized in a localized way even when the noises correlate globally.

B. Convergence and Stopping Criteria

Assume that the optimization problem (14) is feasible, that is, (14) has an optimal solution CL^*. We further assume that the matrix $[C_1 \quad D_{12}]$ has full column rank, and $[B_1; D_{21}]$ has full row rank. In this case, the objective function is strongly convex with respect to CL, and the optimal solution

CL^* is unique. As f and g are closed, proper, and convex, we have strong duality and (14) satisfies the convergence conditions in [20]. From [20], the objective of (14) converges to its optimal value. As the objective function is a continuous function of CL and the optimal solution CL^* is unique, we have primal variable convergence $CL_1^k \to CL^*$ and $CL_2^k \to CL^*$. Note that the rank condition on the objective matrices is a sufficient condition for primal variable convergence. We believe that there exists a less restricted condition for convergence.

The stopping criteria is designed from [20], in which we use $||CL_1^k - CL_2^k||_{\mathcal{H}_2}$ as primal infeasibility and $||CL_2^k - CL_2^{k-1}||_{\mathcal{H}_2}$ as dual infeasibility. The square of these two functions can also be calculated in a localized way as well. The algorithm (15a) - (15c) terminates when $||CL_1^k - CL_2^k||_{\mathcal{H}_2} < \epsilon^{pri}$ and $||CL_2^k - CL_2^{k-1}||_{\mathcal{H}_2} < \epsilon^{dual}$.

When (14) is not feasible, the stopping criteria on primal infeasibility will not be met. We can set a limit on the number of iteration in the ADMM algorithm to resolve this situation. In fact, the convergence result of the ADMM algorithm should be regarded as a localized feasibility check for (14). Interested readers can refer to [24] on the design of spatio-temporal constraint in (5d) to make (14) feasible.

C. Analytic Solution

We now focus on subproblem (15a). This optimization problem is almost identical to (7), except that we add a quadratic penalty on the distance between CL_1 and its reference $CL_2^k - \Lambda^k$. Similar to the LLQR decomposition, we can perform a column-wise decomposition on (15a) and use the notion of localized region to reduce the dimension of the problem. In the reduced dimension, the optimization problem has the form

$$\begin{aligned} \underset{\{x_\ell, u_\ell\}}{\text{minimize}} \; & || \begin{bmatrix} C_{1\ell} & D_{12\ell} \end{bmatrix} \begin{bmatrix} x_\ell \\ u_\ell \end{bmatrix} ||_{\mathcal{H}_2}^2 + \frac{\rho}{2} || \begin{bmatrix} x_\ell \\ u_\ell \end{bmatrix} - \begin{bmatrix} x_\ell^r \\ u_\ell^r \end{bmatrix} ||_{\mathcal{H}_2}^2 \\ \text{subject to} \; & \text{constraints in (9)} \end{aligned} \tag{16}$$

where x_ℓ^r and u_ℓ^r can be derived from $CL_2^k - \Lambda^k$ using localized region. Recall that we impose an FIR constraint on the optimization problem. Problem (16) becomes a finite dimensional affinely constrained quadratic program that can be solved analytically. Specifically, the optimal solution is an affine function of its reference as

$$\begin{bmatrix} \bar{x}_\ell^* \\ \bar{u}_\ell^* \end{bmatrix} = F_a \begin{bmatrix} \bar{x}_\ell^r \\ \bar{u}_\ell^r \end{bmatrix} + F_b \tag{17}$$

where the bar symbol represents the stacked vector of all its spectral components, i.e. $\bar{x}_\ell^r = [x_\ell^r[1]^\top \dots x_\ell^r[T]^\top]^\top$, etc. We only need to compute the constant matrices (F_a, F_b) once, and then update the solution through (17). With this technique, the output feedback problem can be solved almost as fast as the state feedback problem since (17) can be computed almost instantaneously.

D. Localized Synthesis

The column-wise decomposition has an intuitive interpretation on solving (x_ℓ, u_ℓ) for a local disturbance $(\delta x_j, \delta y_j)$. The interpretation of row-wise decomposition is not that intuitive, but can still be explained as follows. For each (x_j, u_j) in the row-wise decomposition, we define $(\delta x_\ell, \delta y_\ell)$ the localized perturbations that can affect (x_j, u_j). The effect of process noise δx_ℓ and sensor noise δy_ℓ are related to each other. To make CL a valid closed loop transfer matrix [19], the effect of $(\delta x_\ell, \delta y_\ell)$ must satisfy the sensing relation described by constraint (5c). The update of this part can also be solved analytically. We use $(\delta x_\ell^r, \delta y_\ell^r)$ to denote the reference solution derived from $CL_1^{k+1} + \Lambda^k$, (E_a, E_b) to denote the update matrices similar to (17). With this interpretation, we summarize the LLQG synthesis algorithm as follows. For simplicity of presentation, we assume that the block sparsity pattern $\mathcal{S}$ is symmetric.

Algorithm 1: Localized LQG Synthesis

Given global plant model (A, B_2, C_2), sparse objective $(C_1, D_{12}, B_1, D_{21})$, and sparse symmetric FIR constraint $\mathcal{S}$;
for *each sub-system* x_j **do**
    Solve the desired transfer function from $(\delta x_j, \delta y_j)$ to (x_ℓ, u_ℓ) and derive update matrices (F_a, F_b);
    Solve the desired transfer function from $(\delta x_\ell, \delta y_\ell)$ to (x_j, u_j) and derive update matrices (E_a, E_b);
if *all the above problems are feasible* **then**
    while *Stopping criteria not met* **do**
        for *each localized region* **do**
            Compute reference (x_ℓ^r, u_ℓ^r);
            Update (x_ℓ, u_ℓ) using (F_a, F_b) and (x_ℓ^r, u_ℓ^r);
            Distribute (x_ℓ, u_ℓ) within its localized region;
            Compute reference $(\delta x_\ell^r, \delta y_\ell^r)$;
            Update $(\delta x_\ell, \delta y_\ell)$ using (E_a, E_b) and $(\delta x_\ell^r, \delta y_\ell^r)$;
            Update Λ in (15c);
            Distribute $(\delta x_\ell, \delta y_\ell)$ and Λ within its localized region;

After solving Algorithm 1, we can implement the controller by (6) in a localized way. Note that the two for loops in Algorithm 1 can be performed in a parallel way. When the model changes locally, we only need to resolve some of the update matrices (F_a, F_b) or (E_a, E_b) in the first part of the algorithm. Then, we run the second part of the iteration, which is fast. This suggests the possibility to do real-time *re-synthesis* of optimal controller for time varying systems.

V. Simulations

In this section, we compare our LLQG controller with centralized and distributed controller through simulation. We then demonstrate the LLQG controller on a randomized heterogeneous system with 9800 states.

A. Comparison

We start with a 20×20 mesh topology representing the interconnection between sub-systems, and drop each edge with probability 0.2. The resulting interconnected topology is shown in Fig. 3. Note that the network may not be strongly connected, but our algorithm still works. The interaction between two neighboring sub-systems is shown in Fig. 1(b). In this example, the number of states, control, and measurement are 800, 400, and 400, respectively. The size is practically too large to compute other distributed optimal control schemes for comparison.

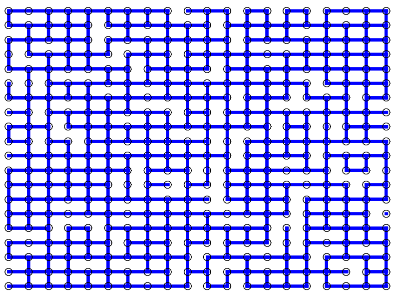


Fig. 3. Interconnected topology for the simulation example

The diagonal entries of A are generated randomly by uniform distribution in $[0.4, 0.8]$. The off-diagonal entries of A are uniformly distributed in $[-0.4, -0.2] \cup [0.2, 0.4]$. The instability of the plant is characterized by the spectral radius of the matrix A, which is 1.1814 in our example. For the objective, we give equal penalty on state deviation and control effort, and the magnitude of process and sensor noise are assumed to be the same. Changing the relative penalty within a range do not affect the result much. Only noisy sensors are required.

For the localized region constraint, we assume that each process noise only affects its neighbors (similar to Fig. 2), and each sensor noise is allowed to affect its two-hop neighbors in closed loop. This means that each sub-system needs to communicate up to its three-hop neighbors during implementation, and use the plant model up to its three-hop neighbors for localized synthesis. We also assume that $u_i[t]$ can access $y_j[\tau]$ 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. For a proper LLQG controller, $u_i[t]$ can access $y_i[t]$. For a strictly proper one, $u_i[t]$ can only access $y_i[t-1]$. We impose the FIR constraint with length $T = 10$ on all closed loop transfer matrices.

We compare the performance of LLQG controller with both proper and strictly proper $\mathcal{H}_2$ optimal controller. The $\mathcal{H}_2$ norm and the total computation time for those control schemes are summarized in Table I. In terms of the $\mathcal{H}_2$ norm, our control scheme have 7.6% and 3.5% degradation compared to an optimal proper and strictly proper controller, respectively. The degradation rate is considered small since the testing plant is highly unstable. In particular, the proper LLQG controller outperforms the strictly proper centralized $\mathcal{H}_2$ one. For total computation time, LLQG takes only about 40% of the time compared to the centralized scheme. It should be noted that we do not utilize any parallel computing technique to calculate the total computation time in Table I. In practice, Algorithm 1 can run in a completely parallel way, and the LLQG controller can be synthesized almost instantaneously.

TABLE I
$\mathcal{H}_2$ NORM AND TOTAL COMPUTATION TIME

	Proper $\mathcal{H}_2$	LLQG	S.P. $\mathcal{H}_2$	S.P. LLQG
$\mathcal{H}_2$ norm	45.35	48.79	49.27	51.00
Comp.Time (s)	134.26	55.41	138.49	55.67

To illustrate the advantages of LLQG, we further compare the LLQG scheme with centralized $\mathcal{H}_2$ in terms of closed loop performance, controller synthesis, and controller implementation in Table II. It can be seen that LLQG is preferable in all aspects, except a slight degradation on the $\mathcal{H}_2$ performance. This degradation is mainly contributed by the FIR constraint. In other words, the locality constraint in this example almost has no effect on the $\mathcal{H}_2$ performance. If we increase the FIR length from 10 to 20, then the $\mathcal{H}_2$ degradation decrease from 3.5% to only 0.2%. This comes with a cost on total computation time, which increases to 231.78 seconds.

TABLE II
COMPARISON BETWEEN CENTRALIZED AND LOCALIZED SCHEME

		Centralized	Localized
Closed Loop	Affected region	Global	2-hop
	Affected time	Long	10 steps
	Normalized $\mathcal{H}_2$	1	1.035
Synthesis	Comp. complexity	$O(n^3)$	$O(n)$
	Comp. time (s)	138.49	55.67
	Time per node (s)	138.49	0.07
	Plant model	Global	3-hop
	Redesign	Offline	Real-time
Implementation	Comm. Speed	Inf	2
	Comm. Range	Global	3-hop

B. Large Example

Lastly, we change the size of the problem and compare the computation time among centralized, distributed, and LLQG scheme. The distributed optimal controller is computed by the method in [7][1], in which we assume the same delay constraint as LLQG. The relation between computation time and the size of problem for different schemes is illustrated in Fig. 4. For LLQG, we plot both the total computation time and the average computation time per sub-system. From Fig. 4, the computation time for the distributed scheme grows rapidly when the size of problem increases. For the centralized one, the slope in the log-log plot in Fig. 4 is 3,

[1]The edge dropping probability is set to zero in this scheme as the method requires strong connectedness of the plant. This does not change computation time much.

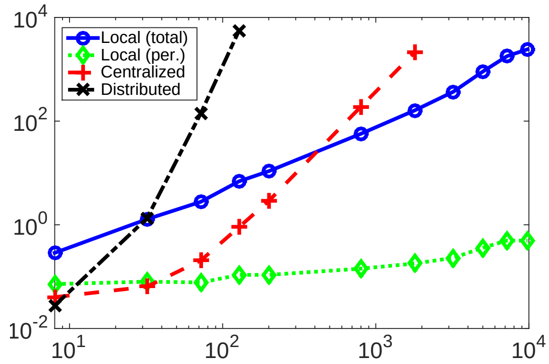


Fig. 4. The horizontal axis represents the number of state, and the vertical axis represents computation time in seconds.

which matches the theoretical complexity $O(n^3)$. The slope for the LLQG (total time) is about 1.4, which is larger than the theoretical value 1. This overhead may be caused by other computational issue such as memory management, and is currently under investigation. For the largest example we have, we can finish the LLQG synthesis for a system with 9800 states in about 40 minutes using a single computer. If the computation is parallelized into all 4900 sub-systems, the synthesis algorithm can be done within a second as indicated by Fig. 4.

VI. Conclusion

In this paper, we introduced the localized LQG (LLQG) synthesis algorithm for large-scale localizable system. This is achieved by combining the localized LQR decomposition technique [18] with distributed optimization. We demonstrated our algorithm on a randomized heterogeneous system with about 10^4 states. As summarized in Table II, the LLQG controller can be implemented and synthesized in a localized way using local plant model, all the while achieving similar transient performance to a centralized optimal one. All these properties are extremely favorable for large-scale interconnected systems.

In the future, we will focus on the *design* issue of a localizable system, including the actuator and sensor placement and the design of the spatio-temporal constraint in LLQG. This may be achieved by the method of regularization for design [24], [25].

Acknowledgements

The authors would like to thank John C. Doyle for his enthusiastic support of this work, and Andrew Lamperski for providing the code for comparison.

References

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

[2] J. N. Tsitsiklis and M. Athans, "On the complexity of decentralized decision making and detection problems," in *IEEE Conference on Decision and Control (CDC)*, 1984.

[3] 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.

[4] 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

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

[6] 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.

[7] ——, "The $\mathcal{H}_2$ control problem for decentralized systems with delays," *arXiv preprint arXiv:1312.7724v2*, 2014.

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

[9] 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.

[10] 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

[11] 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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] N. Motee and A. Jadbabaie, "Approximation methods and spatial interpolation in distributed control systems," in *2009 IEEE American Control Conference (ACC)*, June 2009.

[17] 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.

[18] 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

[19] 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.

[20] 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.

[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] H. Kobayashi and T. Yoshikawa, "Graph-theoretic approach to controllability and localizability of decentralized control systems," *Automatic Control, IEEE Transactions on*, vol. AC-27, no. 5, pp. 1096–1108, 1982.

[23] 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.

[24] Y.-S. Wang, N. Matni, and J. C. Doyle, "Localized LQR control with actuator regularization," in *submitted to 2015 54th IEEE Conference on Decision and Control (CDC)*, 2015.

[25] 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