Localized LQG Optimal Control for Large-Scale Systems

Yuh-Shyang Wang and Nikolai Matni

Abstract— **This paper poses and solves the localized linear quadratic Gaussian (LLQG) optimal control problem. In particular, we show that for large-scale *localizable* systems, that is to say systems for which the closed loop effect of each disturbance can be contained to within a local neighborhood despite communication delays between sub-controllers, the synthesis and implementation of a LLQG optimal controller can be performed in a scalable way. We combine our prior results on the state-feedback version of this problem with the alternating direction method of multipliers (ADMM) algorithm to formulate a synthesis algorithm that can be solved in a distributed fashion, with each subsystem solving a problem of constant dimension independent of the global problem size. The result is a controller synthesis and implementation scheme that can scale to systems of arbitrary dimension, subject to certain conditions on the communication, actuation and sensing schemes holding. Simulations show that for some systems, the LLQG optimal controller can achieve transient performance similar to that of a centralized $\mathcal{H}_2$ optimal controller. We also demonstrate our algorithm on a system with about 10^4 states composed of heterogeneous and dynamically coupled subsystems – here the distributed and centralized optimal controllers cannot be computed.**

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 introduces new challenge when designing a controller: the computational complexity of synthesizing and implementing a controller must be traded off against the closed loop performance that it achieves. 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 attempt 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. Specifically, a distributed optimal control problem is commonly formulated by imposing an information sharing constraint on the controller – the tractability of the resulting problem depends on the relationship between the information sharing constraint and the plant, and certain problems can indeed be NP-hard [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 a means of convexifying the distributed optimal control problem, tractable solutions for various types of distributed constraints and objectives have been developed [5]–[10].

Although the distributed optimal control problem with a QI information sharing constraint is computationally tractable (i.e., convex), both the controller synthesis and implementation scale with the dimension of the full system, and thus are not practical for large systems. In fact, for any plant with a strongly connected topology, an information sharing constraint is QI if and only if the local measurement taken by each sensor is shared among *all* sub-controllers in the distributed system (this follows from the conditions identified in [11]). This then implies that when a plant is strongly connected, an optimal controller $\mathbf{K}$ (the transfer function from measurements $\mathbf{y}$ to control actions $\mathbf{u}$) can only be solved for in a convex manner via the QI Youla parameterization if $\mathbf{K}$ is dense, i.e., if each sub-controller needs to collect the entire measurement $\mathbf{y}$ of the system to compute its control action. This lack of scalability has not gone unnoticed by the community, and techniques based on regularization [12], convex approximation [13], [14], and spatial truncation [15] have been used in hopes of finding a near optimal distributed static feedback controller that is scalable to implement. These methods have been successful in extending the size of systems for which a distributed controller can be computed, but there is still a limit to their scalability as they often rely on an underlying centralized synthesis procedure. Further, it is not clear if these methods can be extended to compute a dynamic controller that incorporates information sharing constraints.

In previous work [16]–[18], we introduced the notion of a localizable system. These are systems for which a distributed controller exists that achieves a closed loop response in which the effect of each disturbance is limited to a local neighborhood (we make this precise in §II). We showed that if a system is localizable, then a controller achieving the desired localized closed loop response can be implemented in a scalable way if we allow the communication of both *measurements* and *controller internal states* between sub-controllers. In particular, this means that each sub-controller i only needs to collect a subset of the measurements and

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.

controller internal states to compute its control action. In addition to a scalable, or localized, implementation, we show that if a system can be localized via state-feedback controller, then the resulting localized linear quadratic regulator (LLQR) optimal controller can be synthesized in a scalable, parallel and localized way [17]. Specifically, we show that the LLQR optimal control problem can be decomposed into several subproblems that can be solved in parallel, each defined in terms of a local subset of the controller and system and admitting an analytic solution. However, this technique does not extend to the output feedback setting [18] because there is an additional constraint that introduces a coupling in the optimization problem.

In this paper, we pose and solve the localized linear quadratic Gaussian (LLQG) problem, i.e., we solve the output feedback localized optimal control problem with an $\mathcal{H}_2$ (LQG) performance metric. We show that by combining the alternating direction method of multipliers (ADMM) with the LLQR decomposition technique, we can synthesize an LLQG optimal controller in a scalable, parallel and localized way. Thus the LLQG controller can be both synthesized and implemented in a scalable way, with the complexity of each task independent of the global system size.

The rest of this paper is structured as follows. In §II we introduce the system model, recall the QI distributed optimal control formulation and comment on its scalability limitations, and recall relevant results from our prior work on localized optimal control. In §III, we recall the LLQR decomposition of the state-feedback problem, as it is a key component of our proposed computational scheme. We then use the LLQR decomposition in combination with the ADMM algorithm to show how the LLQG optimal control problem can be solved in a scalable way in §IV. We end with §V, where we show the effectiveness of our method by synthesizing the LLQG optimal controller for a system with $\sim 10^4$ states composed of dynamically coupled heterogeneous subsystems.

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, and $\mathcal{F}_T$ to denote the space of finite impulse response transfer matrices with horizon T, i.e., $\mathcal{F}_T := \{\mathbf{G} \in \mathcal{RH}_\infty \,|\, , \mathbf{G} = \sum_{t=0}^{T} \frac{1}{z^t} G_t\}$.

II. PROBLEM FORMULATION

We begin by introducing the interconnected system model that we consider in this paper. We then explain why distributed optimal controller synthesis techniques based on the quadratic invariance (QI) framework [3] do not scale to large systems. We end this section by recalling the results of [18] in which we defined the localized optimal control framework.

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, control action u_i, and measurement y_i; and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ 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

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

where $A_{ii}, A_{ij}, B_{ii}, C_{ii}$ are matrices of compatible dimensions, and δ_{x_i} and δ_{y_i} denote state and measurement perturbations, respectively.

B. Quadratic Invariance and Scalability

Designing controllers for interconnected systems (1) can be very challenging for two distinct reasons: (i) in general, the optimal control problem may be intractable [1], and (ii) even if the problem is tractable, the resulting synthesis problem may not scale gracefully to large systems. Although recent work [3] has been very successful in identifying tractable (i.e., convex) classes of such problems, as described in the introduction, there has been less focus on scalability. Before describing our scalable controller synthesis framework, we briefly recall the QI optimal control framework, and show why scalability issues may arise when dealing with large systems. We begin by constructing the global plant

$$\begin{aligned} x[k+1] &= Ax[k] + B_2 u[k] + \delta_x[k] \\ z[k] &= C_1 x[k] + D_{12} u[k], \;\; y[k] = C_2 x[k] + \delta_y[k] \end{aligned} \tag{2}$$

where x, u, y, δ_x, and δ_y are stacked vectors of the subsystem states, controls, measurements, and process and sensor disturbances, respectively, and z is a controlled output. The global plant model (A, B_2, C_2) is constructed such that the dynamics (2) are compatible with those specified in (1). If we let the disturbances be given by $\delta_x = B_1 w$ and $\delta_y = D_{21} w$ for some matrices B_1 and D_{12}, and a disturbance vector w, then equation (2) can be written in transfer function form as

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

where $\mathbf{P}_{ij} = C_i(zI - A)^{-1} B_j + D_{ij}$.

We can then formulate the distributed optimal control problem as

$$\begin{aligned} \underset{\mathbf{K}}{\text{minimize}} \quad & \|\mathbf{P}_{11} + \mathbf{P}_{12}\mathbf{K}(I - \mathbf{P}_{22}\mathbf{K})^{-1}\mathbf{P}_{21}\| \\ \text{subject to} \quad & \mathbf{K} \text{ internally stabilizes } \mathbf{P}, \;\; \mathbf{K} \in \mathcal{C} \cap \mathcal{RH}_\infty \end{aligned} \tag{3}$$

where the subspace constraint $\mathbf{K} \in \mathcal{C}$ enforces information sharing constraints between the sub-controllers. It was shown in [3] that if the constraint set $\mathcal{C}$ is quadratically invariant with respect to $\mathbf{P}_{22}$ then the optimal control problem admits a convex reformulation. Loosely speaking, this condition requires that sub-controllers be able to share information with each other at least as quickly as their control actions propagate through the plant [11].

Assume that for a given constraint $\mathcal{C}$, the optimal controller $\mathbf{K}$ can be computed: the implementation complexity of $\mathbf{K}$ is then determined by the densest row $\mathbf{K}_i$ of $\mathbf{K}$. Specifically, if we consider the control action taken at subsystem i, which is specified by $\mathbf{u}_i = \mathbf{K}_i\mathbf{y}$, then if $\mathbf{K}_i$ is completely dense then subsystem i must collect measurements from every other subsystem $j \in \mathcal{V}$. As the number of subsystems grows large, this then leads to a very complex controller implementation. A natural solution to this problem is to impose sparsity constraints on the controller $\mathbf{K}$ such that each row only has a small number of nonzero terms – in this way each subsystem i only needs to collect a small number of measurements to compute its control action. Unfortunately, this naive approach fails if the dynamics of $\mathbf{P}_{22}$ are strongly connected because in that case, any sparse constraint set $\mathcal{C}$ is not QI with respect to $\mathbf{P}_{22}$. Although recent methods based on convex relaxations [13] can be used to solve certain cases of the non-convex optimal control problem (3) with sparse constraint set $\mathcal{C}$, the underlying synthesis optimization problem is itself still large-scale and does not admit a scalable reformulation. The need to address scalability, both in the synthesis and implementation of a controller, is the driving motivation behind the localized optimal control framework, which we recall in the next subsection.

C. Localized Distributed Control

Our previous work on localized optimal control [16]–[18] is built on the observation that the controller need not be implemented as a map $\mathbf{K}$ from observations $\mathbf{y}$ to control actions $\mathbf{u}$, but rather admits a parameterization in terms of four closed loop transfer matrices $(\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L})$ that define the closed loop maps from $\boldsymbol{\delta_x}$ and $\boldsymbol{\delta_y}$ to the state $\mathbf{x}$ and control action $\mathbf{u}$ (we make this precise in what follows). We show that by working directly with these transfer matrices, suitably structured controllers and controller synthesis problems can be defined that address the issues of scalable synthesis and implementation described above. In [18] we show that for a system described by (2), there exists a controller such that the closed loop transfer matrices $(\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L})$ satisfy

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

if and only if $(z\mathbf{R}, z\mathbf{M}, z\mathbf{N}, \mathbf{L}) \in \mathcal{RH}_\infty$ and

$$\begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} = \begin{bmatrix} I & 0 \end{bmatrix} \tag{5a}$$

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

The *localized* distributed optimal control problem is then posed as

$$\underset{\{\mathbf{R},\mathbf{M},\mathbf{N},\mathbf{L}\}}{\text{minimize}} \quad \| \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix} \begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} \| \tag{6a}$$

$$\text{subject to (5a) and (5b)}, \ \begin{bmatrix} z\mathbf{R} & z\mathbf{N} \\ z\mathbf{M} & \mathbf{L} \end{bmatrix} \in \mathcal{C} \cap \mathcal{L} \cap \mathcal{F}_T \tag{6b}$$

where $\mathcal{C}$ is a subspace encoding the information sharing constraints of the distributed controller, $\mathcal{L}$ is a *localized* constraint and $\mathcal{F}_T$ restricts the optimization variables to have finite impulse responses of horizon T. 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 delving into what these subspaces encode and how to design them, we recall the following theorem from [18].

Theorem 1: Suppose that a set of transfer matrices $(\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L})$ satisfy constraints (5a) and (5b). Then a controller yielding the closed loop response (4) can be implemented as

$$\boldsymbol{\beta} = \tfrac{1}{z}\tilde{\mathbf{R}}^+\boldsymbol{\beta} + \tfrac{1}{z}\tilde{\mathbf{N}}\mathbf{y}, \ \mathbf{u} = \tilde{\mathbf{M}}\boldsymbol{\beta} + \mathbf{L}\mathbf{y} \tag{7}$$

where $\tilde{\mathbf{R}}^+ = z(I - z\mathbf{R})$, $\tilde{\mathbf{N}} = -z\mathbf{N}$, $\tilde{\mathbf{M}} = z\mathbf{M}$, $\mathbf{L}$ are stable proper transfer matrices and $\boldsymbol{\beta}$ is the controller's internal state. Further, the implementation (7) is internally stabilizing.

Remark 1: Whereas the QI framework requires a transformation between the optimization variable and the controller (in particular the Youla parameter $\mathbf{Q}$ must be mapped back to the controller $\mathbf{K}$ via a linear fractional transform), we directly implement the controller in terms of the optimization variables $(\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L})$. In doing so, we are able to avoid the issues of non-convexity that are present in the QI framework – however, it is possible to impose constraints that lead to an infeasible problem, and thus our results are still consistent with established results on the hardness of distributed control subject to non-classical information patterns [1], [2].

Theorem 1 thus shows how the sparsity of the closed loop transfer matrices $(\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L})$ translates into the implementation complexity of a controller as in (7). If the rows of these transfer matrices are suitably sparse, then each sub-controller only needs to collect a small number of measurements and controller internal states to compute its control law. 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 $\text{dist}\,(j \to i)$ 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 $\text{dist}\,(i \to j) \leq d$ for all $i, j \in \mathscr{C}$. We then define the d-outgoing set of subsystem j as $\text{Out}_j(d) := \{i | \text{dist}(j \to i) \leq d\}$, and the d-incoming set of subsystem j as $\text{In}_j(d) := \{i | \text{dist}(i \to j) \leq d\}$ – by definition, both of these sets are of size d. Our approach to making the controller synthesis task specified in optimization problem (6) scalable is to confine, or *localize* the effects of the process and sensor disturbances 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 $\text{In}_j(d)$.

Example 1: For a system (1) with interaction graph illustrated in Fig. 1, the 2-incoming and 2-outgoing sets of subsystem 5 are given by $\text{In}_5(2) = \{2, 3, 4, 5\}$ and $\text{Out}_5(2) = \{5, 6, 7, 8, 9, 10\}$, respectively.

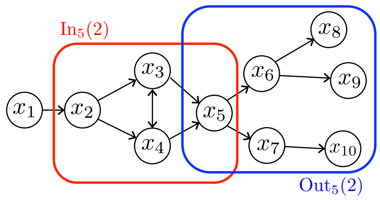


Fig. 1: Illustration of the 2-incoming and 2-outgoing sets of subsystem 5.

With this approach in mind, we say that the transfer matrix $\mathbf{R}$ mapping the state disturbance $\boldsymbol{\delta}_x$ to the state $\mathbf{x}$, as defined in (4), is d-localized if its impulse response can be appropriately covered by d-outgoing sets. In particular, if we let $\mathbf{R}_{ij}$ denote the transfer function from the perturbation $\boldsymbol{\delta}_{x_j}$ at sub-system j to the state $\mathbf{x}_i$ at sub-system i, then the map $\mathbf{R}$ is d-localized if and only if for every subsystem j, $\mathbf{R}_{ij} = 0$ for all $i \notin \text{Out}_j(d)$. In words, this says that the transfer matrix $\mathbf{R}$ is d-localized if and only if the effect of each disturbance is contained, or localized, to within a region of size d. This definition can be extended to the remaining transfer matrices $\mathbf{N}$, $\mathbf{M}$, and $\mathbf{L}$ of (4) in a natural way. The closed loop response (4) of the system is said to be d-localized if its constituent components $(\mathbf{R}, \mathbf{N}, \mathbf{M}, \mathbf{L})$ are as well. Finally, the subspace $\mathcal{L}$ in (6b) imposes a d-localized constraint if it constrains its elements to be d-localized closed loop responses.

With these definitions and results in hand, our approach to synthesizing a controller that is scalable to implement is straightforward: we choose the subspace $\mathcal{L}$ to be a d-localized constraint and select a horizon T for the FIR subspace $\mathcal{F}_T$. Whereas the first constraint ensures that the transfer matrices solving optimization problem (6) are d-localized, the second constraint is simply imposed to make the optimization problem finite dimensional. The resulting optimization problem (6) is easily seen to be convex as the constraints are all affine: if it is feasible, then the resulting closed loop transfer matrices $(\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L})$ can be used to implement a controller as specified by Theorem 1 (note that the modified transfer matrices $(\tilde{\mathbf{R}}, \tilde{\mathbf{M}}, \tilde{\mathbf{N}})$ are also d-localized). Further, since the effect of each disturbance is localized to a region of size d (i.e., since the closed loop response is d-localized), each sub-controller only needs to collect data from other subsystems that are at most a distance d away. In particular, this means that the row sparsity of the transfer matrices $(\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L})$ is defined by the d-incoming sets of the system, and thus the control action $\mathbf{u}_i$ and internal state $\boldsymbol{\beta}_i$ of sub-controller i can be computed by collecting the data $(\mathbf{y}_j, \boldsymbol{\beta}_j)$ from subsystems j in the d-incoming set $\text{In}_i(d)$. This discussion shows that a controller synthesized via optimization problem (6) and implemented according to Theorem 1 is scalable to implement, as the amount of information collected at a given subsystem is specified by the d-localized constraint $\mathcal{L}$, and hence is independent of the number of subsystems n.

Before proceeding, we note that although optimization problem (6) is convex for any affine constraint set $\mathcal{L}$, it is not necessarily feasible. This is because the subsystems are dynamically coupled to each other (as per (1)), and hence it may not always be possible to localize the effect of a disturbance affecting subsystem j to a pre-specified d-outgoing set $\text{Out}_j(d)$. Intuitively, optimization problem (6) is feasible when the communication constraint $\mathcal{C}$ is such that information about a disturbance can be transmitted faster than the disturbance propagates through the plant. In this way, the necessary sub-controllers can take action *before* a disturbance perturbs their subsystem and stop it from propagating to the rest of the system. Implicit in this argument are the assumptions that information can be shared quickly between sub-controllers, and that sensors and actuators are strategically placed (these assumptions are discussed in detail in [17]–[19]).

Based on this intuition, we provide simple rules for how to select the size d of the localized subspace constraint $\mathcal{L}$, the horizon T of the FIR subspace constraint $\mathcal{F}_T$, as well as a principled approach to designing the actuation scheme of the system (i.e., the matrix B_2) in our companion paper [19]. For the remainder of the paper, we assume that the horizon T has been set, that the subspace constraint $\mathcal{L}$ is d-localized, and that the communication, actuation and sensing architecture of the system is such that optimization problem (6) is feasible.

D. Problem Statement

We have thus far argued that localized controllers are scalable to implement – what remains to be shown is that they are also scalable to synthesize. The goal of this paper is to show that optimization problem (6) can be solved in a scalable manner when the objective function (6a) is taken to be the $\mathcal{H}_2$ (LQG) norm. Our approach is to exploit the structure of the d-localized subspace constraint $\mathcal{L}$ and the interconnection graph underlying the dynamics (2) to decompose the large-scale global optimization problem (6) into easily solved sub-problems of moderate size. In what follows we present an ADMM based algorithm for solving problem (6) that requires each subsystem j to solve a sub-problem defined only by the state-space parameters of subsystems contained in its d-outgoing and d-incoming sets $\text{Out}_j(d)$ and $\text{In}_j(d)$, respectively. In addition, the computation performed at each subsystem can be done parallel: thus the computational complexity, memory usage and parallel computation time needed at each subsystem are only a function of the size of the d-incoming and d-outgoing sets of the system.

III. LLQR Decomposition

In this section, we recall the LLQR decomposition technique [17], which is a scalable algorithm used for solving the state-feedback version of the localized optimal control problem (6).

A. Column-wise Decomposition

Assume that the plant (2) corresponds to a state-feedback model, i.e., that $C_2 = I$ (full sensing) and $D_{21} = 0$ (no sensor noise), and that $B_1 = I$ (uncorrelated process noise). The LLQR problem [17] then follows as a special case of optimization problem (6), and is given by

$$\begin{aligned} \underset{\{\mathbf{R},\mathbf{M}\}}{\text{minimize}} \quad & \| \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} \|_{\mathcal{H}_2}^2 \\ \text{subject to} \quad & \begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix} = I \\ & \begin{bmatrix} z\mathbf{R}^\top & z\mathbf{M}^\top \end{bmatrix}^\top \in \mathcal{C} \cap \mathcal{L} \cap \mathcal{F}_T. \end{aligned} \tag{8}$$

As shown in [17], optimization problem (8) admits a column-wise decomposition. This decomposition follows directly from the LTI property of the system and the decomposability of the LQR cost for uncorrelated process noise.[1] Specifically, the objective function of (8) can be written as

$$\sum_{j=1}^{n} \| \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R}_j \\ \mathbf{M}_j \end{bmatrix} \|_{\mathcal{H}_2}^2 \tag{9}$$

where $\mathbf{R}_j$ and $\mathbf{M}_j$ denote the jth block column of $\mathbf{R}$ and $\mathbf{M}$, which correspond to the closed loop maps from the process noise $\boldsymbol{\delta}_{x_j}$ at subsystem j to the global state $\mathbf{x}$ and control action $\mathbf{u}$, respectively. Recall that if $\mathcal{L}$ is a d-localized constraint, then the ith block-row of this block-column is nonzero only if $i \in \text{Out}_j(d)$, i.e., if subsystem i is in the d-outgoing set of subsystem j. Using a similar argument, we can also decompose the constraints of (8) in a block-column wise manner. Thus, if we let $\mathcal{S}_j$ denote the constraint imposed on block-column j by a subspace $\mathcal{S}$, we can solve the collection of subproblems

$$\begin{aligned} \underset{\{\mathbf{R}_j,\mathbf{M}_j\}}{\text{minimize}} \quad & \| \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R}_j \\ \mathbf{M}_j \end{bmatrix} \|_{\mathcal{H}_2}^2 \\ \text{subject to} \quad & \begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{M} \end{bmatrix}_j = [I]_j \\ & \begin{bmatrix} z\mathbf{R}_j^\top & z\mathbf{M}_j^\top \end{bmatrix}^\top \in \mathcal{C}_j \cap \mathcal{L}_j \cap \mathcal{F}_T. \end{aligned} \tag{10}$$

for $j = 1, \dots, n$ instead of solving the global optimization problem (8). We already alluded to the benefit of this decomposition above: for a d-localized constraint $\mathcal{L}$, the nonzero block-rows of each $\mathbf{R}_j$ and $\mathbf{M}_j$ is at most of size d as specified by $\text{Out}_j(d)$, the d-outgoing set at subsystem j, and thus we can greatly reduce the number of the optimization variables in problem (10) by discarding those constrained to be zero. Further, because of the decomposability (9) of the $\mathcal{H}_2$ norm and the assumption of uncorrelated process noise, we can independently (and hence in parallel) optimize the closed loop response to each local disturbance $\boldsymbol{\delta}_{x_j}$ using the localized subproblems (10), even when their outgoing sets $\text{Out}_j(d)$ overlap.

In order to specify the subproblems (10) in terms of their reduced optimization variables (which will be needed to specify our ADMM based algorithm in the next section), we introduce some further notation. 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 $i \in \text{Out}_j(d+1)$,[2] 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. Finally let $I_{o_{j,d}}$ be the corresponding submatrix of the identity. Each LLQR sub-problem (10) can then be written as

$$\begin{aligned} \underset{\{\mathbf{R}_{o_{j,d}},\mathbf{M}_{o_{j,d}}\}}{\text{minimize}} \; & \| \begin{bmatrix} C_{1o_{j,d}} & D_{12o_{j,d}} \end{bmatrix} \begin{bmatrix} \mathbf{R}_{o_{j,d}} \\ \mathbf{M}_{o_{j,d}} \end{bmatrix} \|_{\mathcal{H}_2}^2 \\ \text{s.t.} \; & \begin{bmatrix} zI - A_{o_{j,d}} & -B_{2o_{j,d}} \end{bmatrix} \begin{bmatrix} \mathbf{R}_{o_{j,d}} \\ \mathbf{M}_{o_{j,d}} \end{bmatrix} = I_{o_{j,d}} \\ & \begin{bmatrix} z\mathbf{R}_{o_{j,d}}^\top & z\mathbf{M}_{o_{j,d}}^\top \end{bmatrix}^\top \in \mathcal{C}_{o_{j,d}} \cap \mathcal{L}_{o_{j,d}} \cap \mathcal{F}_T. \end{aligned} \tag{11}$$

where we use $\mathcal{S}_{o_{j,d}}$ to denote the projection of a subspace $\mathcal{S}_j$ onto the nonzero block-rows specified by $\text{Out}_j(d)$.

IV. Localized LQG Synthesis

The LLQR decomposition technique cannot be applied to optimization problem (6) for plants (2) corresponding to output feedback problems. This is because the constraints (5a) and (5b) admit incompatible decompositions: constraint (5a) can be decomposed block-column wise, whereas constraint (5b) can be decomposed block-row wise, introducing a coupling between all optimization variables. The ADMM has proven very useful in "breaking" such coupling between optimization variables, allowing for large-scale problems to be decomposed and solved efficiently. Our approach to developing a scalable solution to the localized optimal control problem (6) is to combine the ADMM technique with the LLQR decomposition introduced in the previous section.

To reduce notational clutter, we assume that $B_1 = \begin{bmatrix} I & 0 \end{bmatrix}$ and $D_{21} = \begin{bmatrix} 0 & \sigma_y I \end{bmatrix}$, where σ_y is the relative magnitude between process disturbance and sensor disturbance.[3]Using these values for B_1 and D_{21}, and the $\mathcal{H}_2$ norm in the localized optimal control problem (6) yields the localized LQG (LLQG) optimal control problem

$$\underset{\{\mathbf{R},\mathbf{M},\mathbf{N},\mathbf{L}\}}{\text{minimize}} \; \| \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R} & \sigma_y \mathbf{N} \\ \mathbf{M} & \sigma_y \mathbf{L} \end{bmatrix} \|_{\mathcal{H}_2}^2 \tag{12a}$$

$$\text{subject to (5a) and (5b)} \tag{12b}$$

$$\begin{bmatrix} z\mathbf{R} & z\mathbf{N} \\ z\mathbf{M} & \mathbf{L} \end{bmatrix} \in \mathcal{C} \cap \mathcal{L} \cap \mathcal{F}_T. \tag{12c}$$

We continue to assume that the subspace $\mathcal{L}$ is a d-localized constraint, ensuring that the resulting optimal controller admits a scalable implementation, as described in §II-C. We now make a series of observations that motivate the use of the ADMM algorithm to solve the LLQG problem (12). First, notice that if we remove constraint (5b) from problem

[1]These results extend in a natural way to when B_1 is a diagonal or block-diagonal matrix.

[2]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.

[3]The methods in this section extend in a natural way to diagonal and block-diagonal matrices B_1 and D_{21}.

(12), then the resulting optimization problem admits a block column-wise LLQR decomposition, which as described in the previous section allows for the global problem to be decomposed into subproblems of size defined by that of the d-outgoing sets of the subsystems. Through a dual argument, we can show that verifying the feasibility of constraint (5b) can done block-row at a time, resulting in a feasibility problem that admits a block row-wise LLQR decomposition, once again allowing for the global problem to be decomposed into easily solved subproblems. In order to exploit the decomposition properties of each of these modified problems, we leverage the standard ADMM technique of shifting the coupling from the difficult to enforce constraints (5a) and (5b) to a simple equality constraint through the introduction of a redundant variable: we make this approach precise in what follows. We use

$$\Psi_i = \begin{bmatrix} \mathbf{R}^i & \mathbf{N}^i \\ \mathbf{M}^i & \mathbf{L}^i \end{bmatrix}$$

to denote the ith copy of the closed loop transfer matrix that we are solving for. Following [20], we define the extended-real-value functions $f(\Psi_1)$ and $g(\Psi_2)$ as

$$\begin{array}{rcll} f(\Psi_1) & = & \{(12a) & \text{if } (5a), (12c), \infty \text{ otherwise}\} \\ g(\Psi_2) & = & \{0 & \text{if } (5b), (12c), \infty \text{ otherwise}\}. \end{array} \tag{13}$$

Using these definitions, we can rewrite the LLQG optimization problem (12) as

$$\underset{\{\Psi_1, \Psi_2\}}{\text{minimize}} \quad f(\Psi_1) + g(\Psi_2) \text{ subject to } \Psi_1 = \Psi_2. \tag{14}$$

The form of optimization problem (14) is precisely that needed by the ADMM approach [20], and can be solved via the iterations

$$\Psi_1^{k+1} = \underset{\Psi_1}{\text{argmin}} \left(f(\Psi_1) + \frac{\rho}{2} ||\Psi_1 - \Psi_2^k + \Lambda^k||_{\mathcal{H}_2}^2 \right) \tag{15a}$$

$$\Psi_2^{k+1} = \underset{\Psi_2}{\text{argmin}} \left(g(\Psi_2) + \frac{\rho}{2} ||\Psi_2 - \Psi_1^{k+1} - \Lambda^k||_{\mathcal{H}_2}^2 \right) \tag{15b}$$

$$\Lambda^{k+1} = \Lambda^k + \Psi_1^{k+1} - \Psi_2^{k+1}. \tag{15c}$$

Recall that each closed loop response Ψ_i is constrained to lie in the FIR subspace $\mathcal{F}_T$, and hence is a finite dimensional variable: it follows that each of the problems specified by the ADMM algorithm (15) can be formulated as finite dimensional optimization problems by associating the FIR transfer matrices with their matrix representations. We now focus on the problem specifying the Ψ_1^k iterates, which can be written as

$$\begin{aligned} \underset{\{\mathbf{R}^1, \mathbf{M}^1, \mathbf{N}^1, \mathbf{L}^1\}}{\text{minimize}} \; & \| \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{R}^1 \\ \mathbf{M}^1 \end{bmatrix} \|_{\mathcal{H}_2}^2 \\ & + \sigma_y^2 \| \begin{bmatrix} C_1 & D_{12} \end{bmatrix} \begin{bmatrix} \mathbf{N}^1 \\ \mathbf{L}^1 \end{bmatrix} \|_{\mathcal{H}_2}^2 + \tfrac{\rho}{2} ||\Psi_1 - \Psi_2^k + \Lambda^k||_{\mathcal{H}_2}^2 \\ \text{subject to } & \begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} \mathbf{R}^1 \\ \mathbf{M}^1 \end{bmatrix} = I \\ & \begin{bmatrix} zI - A & -B_2 \end{bmatrix} \begin{bmatrix} \mathbf{N}^1 \\ \mathbf{L}^1 \end{bmatrix} = 0 \\ & \begin{bmatrix} z\mathbf{R}^1 & z\mathbf{N}^1 \\ z\mathbf{M}^1 & \mathbf{L}^1 \end{bmatrix} \in \mathcal{C} \cap \mathcal{L} \cap \mathcal{F}_T. \end{aligned} \tag{16}$$

From the form of this problem, it is apparent that an analogous argument to that presented in §III applies and that a block column-wise LLQR decomposition can be applied to the objective function and constraints, allowing for subproblems of scale d to be solved. Similarly, subproblem (15b) admits a block row-wise LLQR decomposition, and the Lagrange multiplier update equation (15c) decomposes element-wise. Thus if the ADMM weight ρ is shared between subsystems prior to the synthesis procedure, the optimization problems specifying the ADMM algorithm (15) decompose into subproblems specified by the d-outgoing and d-incoming sets of the system.

Next we show that the problems specifying the iterates Ψ_i^k can be solved in closed form allowing for the update equations (15a) and (15b) to be implemented via matrix multiplication. We end the section with a discussion of conditions guaranteeing the convergence of the iterates Ψ_i^k to the optimal solution to the LLQG problem (12).

Remark 2: The ADMM approach specified in (15) can be used with other objective functions that admit a block column-wise decomposition and/or block row-wise decomposition. An interesting special case is that we can solve problem (12) for arbitrary B_1 and D_{21} if $[C_1 \quad D_{12}]$ is diagonal – in particular, this means that an LLQG controller can be synthesized in a scalable way using our proposed algorithm if the process and sensor noise are globally correlated, so long as the subsystem's performance objectives are decoupled.

A. Analytic Solution

We now focus on optimization problem (16), which specifies the iterates Ψ_1^k. Following the LLQR method described in §III, we perform a block-column-wise decomposition of the objective and constraints of (16), and exploit the d-localized structure of the system to reduce the dimensionality of each resulting subproblem. Specifically, for each disturbance $\boldsymbol{\delta}_{x_j}$ or $\boldsymbol{\delta}_{y_j}$ at subsystem j, we solve an optimization of the same form as (16) except with all decision variables, state-space parameters and constraints restricted to the d-outgoing set of subsystem j, $o_{j,d} := \text{Out}_j(d)$. The result is an optimization problem similar to (11) with solution $\{\mathbf{R}^1_{o_{j,d}}, \mathbf{M}^1_{o_{j,d}}, \mathbf{N}^1_{o_{j,d}}, \mathbf{L}^1_{o_{j,d}}\}$. We also note that optimization problem (11) and the dimensionality reduced version of optimization problem (16) are least-squares problems subject to affine constraints. Consequently, the optimal solution is specified as an affine function of the problem data $\left(\Psi_2^k\right)_{o_{j,d}}$ and $\Lambda^k_{o_{j,d}}$, and can be written

$$(\Psi_1)^{k+1}_{o_{j,d}} = F^a_{o_{j,d}} \left(\left(\Psi_2^k\right)_{o_{j,d}}, \Lambda^k_{o_{j,d}} \right) + F^b_{o_{j,d}}, \tag{17}$$

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 $\|G\|_F^2 = \|\text{vec}(G)\|_2^2$. In particular, the terms $(F^a_{o_{j,d}}, F^b_{o_{j,d}})$ only need to be computed once, after which the updates to the iterates Ψ_1^k can be performed via equation (17). Note that this procedure can be performed in parallel at each subsystem j.

An analogous argument allows us to solve the Ψ_2^k iterate update equation (15b) by applying a block row-wise decomposition. For each subsystem j, let $\mathrm{in}_{j,d} = \mathrm{In}_j(d)$ and define $(\delta_{\mathrm{in}_{j,d}})$ to be the collection of process and sensor disturbances that affect the d-incoming set of subsystem j. The decomposition and dimensionality reduction of this subproblem then proceeds in a manner dual to that of (16) using a block row-wise decomposition and a reduction to subproblems of dimension specified by the d-incoming sets of this system. Once again, the resulting dimensionality reduced problems are least-squares problems subject to affine constraints, and their optimal solutions are affine functions of the problem data $\left(\Psi_1^{k+1}\right)_{\mathrm{in}_{j,d}}$ and $\Lambda^k_{\mathrm{in}_{j,d}}$, and can be written

$$(\Psi_2)^{k+1}_{o_{j,d}} = E^a_{\mathrm{in}_{j,d}}\left(\left(\Psi_1^{k+1}\right)_{\mathrm{in}_{j,d}}, \Lambda^k_{\mathrm{in}_{j,d}}\right) + E^b_{\mathrm{in}_{j,d}}, \quad (18)$$

for a suitable linear map $E^a_{\mathrm{in}_{j,d}}$ and affine term $E^b_{\mathrm{in}_{j,d}}$. This procedure can also be done in parallel at each subsystem j.

Thus using this approach to solving the iterate updates (15a) and (15b), the LLQG optimization problem (12) can be solved nearly as quickly as the state-feedback problem, as the update equations (17) and (18) require first solving a least-squares problem defined on the d-incoming and d-outgoing sets of the system and then using matrix multiplication. Once the optimal controller is solved for using the method described above, it can then be implemented according to (7): as we argued in §II, imposing that the constituent transfer matrices $(\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L})$ be d-localized implies that the controller admits a localized implementation. Thus we have met our goal of designing a controller that can be both synthesized and implemented in a localized, and hence scalable, manner.

An added benefit of the localized optimal control framework is the ability to perform real-time re-synthesis of optimal controllers. In particular, suppose that the dynamics (1) describing the dynamics of a collection $\mathscr{C}$ of subsections change – in order to suitably update the LLQG optimal controller, only the components of the closed loop transfer matrices $(\mathbf{R}, \mathbf{M}, \mathbf{N}, \mathbf{L})$ corresponding to the response of subsystems j satisfying $\mathrm{In}_j(d) \cap \mathscr{C} \neq \emptyset$ or $\mathrm{Out}_j(d) \cap \mathscr{C} \neq \emptyset$ need to be updated according to equations (17) and (18).

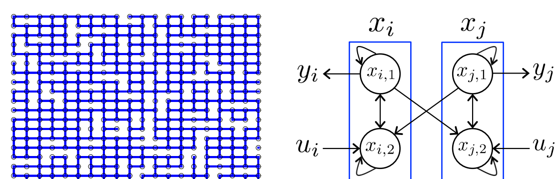
Convergence and Stopping Criteria: Assume that the optimization problem (14) is feasible, and let Ψ^* be an optimal solution. Further assume that the matrix $[C_1 \quad D_{12}]$ has full column rank, and $[B_1; D_{21}]$ has full row rank. In this case, the objective function is strongly convex with respect to Ψ, and hence any optimal solution Ψ^* is the unique optimal solution. As the extended functions f and g specified in (13) are closed, proper, and convex we have that strong duality holds and that optimization problem (14) satisfies the convergence conditions state in [20]. From [20], the objective of (14) converges to its optimal value. As the objective function is a continuous function of Ψ and the optimal solution Ψ^* is unique it follows that the primal variable iterates converge to Ψ^*, i.e., $\Psi_1^k \to \Psi^*$ and $\Psi_2^k \to \Psi^*$. Note that the rank condition on the objective function matrices is only a sufficient condition for primal variable convergence, and we believe that less restrictive conditions can be derived. We refer the reader to [20] for how to select stopping criteria for algorithm (15) – this stopping criteria can be further used as a scalable method of verifying the feasibility of optimization problem (14) for a specific choice of localized subspace constraint $\mathcal{L}$ and FIR horizon T. As mentioned previously, we defer a discussion of how to choose these parameters to our companion paper [19].

V. Simulations

Comparison to Traditional Controllers: We begin with a 20×20 mesh topology, which encodes the interconnection between subsystems, and then drop each edge with a probability of 0.2. The resulting interconnection topology is shown in Fig. 2(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 sub-systems is shown in Fig. 2(b), in which state $x_{i,1}$ can affect state $x_{j,2}$ if i and j are neighbors. This model is inspired from the second-order dynamics of a power network or mechanical system. Specifically, consider the 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 state θ_i. We let $x_i := [\theta_i \quad \dot{\theta}_i]^\top$, allow each sub-controller to measure θ_i, and use $e^{A\Delta t} \approx I + A\Delta t$ to obtain a discrete-time plant with interactions between neighboring subsystems described by Fig. 2(b).



(a) Interconnected topology (b) Interaction between neighboring sub-systems

Fig. 2: Simulation example interaction graph.

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 1.1814 in the example that we present. For the objective, we assign equal penalty to the state deviation and control effort, and the magnitudes of the process and sensor disturbances are assumed to be the same.

We construct the localized constraint $\mathcal{L}$ by imposing that the effect of each process disturbance δ_{x_j} be limited to the 2-outgoing set $\mathrm{Out}_j(2)$, and that the effect of each sensor disturbance δ_{y_j} be limited to the 3-outgoing set $\mathrm{Out}_j(3)$. This implies that each subsystem j needs to collect measurements y_i and controller states β_i from sub-controllers i in $\mathrm{In}_j(3)$ to compute its control action, and use the restricted plant model specified by $\mathrm{In}_j(3) \cup \mathrm{Out}_j(3)$ to solve and implement the update equations (17) and (18). We assume that the communication constraint $\mathcal{C}$ is such that at time t, controller i can receive $(y_j[\tau], \beta_j[\tau])$ for all $\tau \leq t - k$ if $\mathrm{dist}\,(j \to i) = k$. The interaction between subsystems

	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

TABLE I: $\mathcal{H}_2$ Norm and Total Computation Time

illustrated in Fig. 2(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. We set the FIR horizon to $T = 10$ for all closed loop transfer matrices: as we show below, this choice leads to near-optimal performance relative to the centralized optimal controller.

We compare the performance of the LLQG controller to both proper and strictly proper $\mathcal{H}_2$ centralized optimal controllers. In this example, the dimension of the state, control action, and measurement vectors are 800, 400, and 400, respectively. Due to the size of the system, it is impossible to compute a distributed optimal controller using the methods described in [6], but we note that the performance of this distributed controller will be no better than that of the centralized controllers to which we compare our scheme.

The $\mathcal{H}_2$ norm and the total computation time for these control schemes are summarized in Table I. As can be seen, the LLQG controller achieves a closed loop $\mathcal{H}_2$ norm that is 7.6% and 3.5% worse than that achieved by the proper and strictly proper centralized optimal controller, respectively. Given that the open-loop plant is unstable, this degradation in performance is quite small, especially given the computational advantages of using the LLQG scheme: the LLQG controller synthesis procedure is completed in approximately 40% of the time needed to compute the centralized controller. It should be noted that we do not utilize any parallel computation to calculate the computation times in Table I. In practice, update equations (17) and (18) can be run in parallel: thus the LLQG controller can be synthesized nearly instantaneously.

To further illustrate the advantages of the LLQG optimal controller, we compare the strictly proper LLQG and $\mathcal{H}_2$ centralized optimal controllers in terms of closed loop performance, controller synthesis complexity, and controller implementation complexity in Table II. It can be seen that the LLQG optimal controller is vastly preferable in all aspects, except for a slight degradation in the closed-loop performance. This degradation is mainly due to the length of the FIR horizon T. In particular, the localized constraint $\mathcal{L}$ in this example has almost no effect on the closed loop performance. If we increase the FIR horizon T from 10 to 20, then the performance degradation decreases from 3.5% to only 0.2%. This however does come at the cost of a larger computation time which increases to 231.78 seconds (note that this time is for a serial computation, and can be reduced if the iterate updates are implemented in parallel).

Large Scale Example: We now allow the size of the problem to vary and compare the computation time needed to synthesize a centralized, distributed, and LLQG optimal controller. The distributed optimal controller is computed using the methods described in [6], in which we assume the same communication constraints $\mathcal{C}$ as LLQG. The empirical

		Centralized	Localized
Closed Loop	Affected region	Global	$\text{Out}_j(3)$
	Affected time	Long	$T = 10$ 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	$\text{Out}_j(3) \cup \text{In}_j(3)$
	Redesign	Offline	Real-time
Implement.	Comm. Speed	∞	2x
	Comm. Range	Global	$\text{In}_j(3)$

TABLE II: Comparison Between Centralized and Localized Control

relationship obtained between computation time and problem size for the different control schemes is illustrated in Fig. 3. For the LLQG controller, we plot both the total computation time and the average computation time per subsystem. As can be seen in Fig. 3, the computation time needed for the distributed controller grows rapidly when the size of problem increases. For the largest example that we computed, we are able to synthesize an LLQG optimal controller for a system with 9800 states in about 40 minutes using a laptop. If the computation were to be parallelized across the 4900 subsystems (as would be done in a practical situation), the synthesis procedure can be performed in under a second.

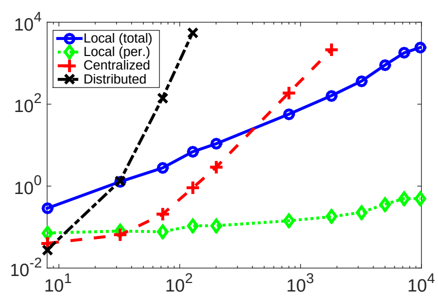


Fig. 3: The horizontal axis denotes the number of states, and the vertical axis denotes computation time in seconds.

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 dec. making and detection problems," in *IEEE Conf. on Dec. and Control (CDC)*, 1984.

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

[4] L. Lessard and S. Lall, "Convexity of decentralized controller synthesis," *submitted to IEEE TAC*, 2014.

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

[6] A. Lamperski and J. C. Doyle, "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:1306.0036v2*, 2014.

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

[9] N. Matni, "Distributed control subject to delays satisfying an $\mathcal{H}_\infty$ norm bound," in *53rd IEEE Conf. on Dec. and Control (CDC)*, 2014.

[10] L. Lessard, "State-space solution to a minimum-entropy $\mathcal{H}_\infty$-optimal control problem with a nested information constraint," in *53rd IEEE Conf. on Dec. and Control (CDC)*, 2014.

[11] 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.

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

[13] G. Fazelnia, R. Madani, and J. Lavaei, "Convex relaxation for optimal distributed control problem," in *2014 53rd IEEE Conf. on Dec. and Control (CDC)*, 2014.

[14] Y. Wang, J. Lopez, and M. Sznaier, "Sparse static output feedback controller design via convex optimization," in *2014 53rd IEEE Conf. on Dec. and Control (CDC)*, 2014.

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

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

[17] Y.-S. Wang, N. Matni, and J. C. Doyle, "Localized LQR optimal control," in *53rd IEEE Conf. on Dec. and Control (CDC)*, 2014.

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

[19] Y.-S. Wang, N. Matni, and J. C. Doyle, "Localized LQR control with actuator regularization," in *submitted to ACC 2016*.

[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," *FoTM*, vol. 3, no. 1, pp. 1–122, 2011.

[21] Y.-S. Wang and N. Matni, "Localized lqg optimal control for large-scale localizable systems," *Extended Report*, 2015. [Online]. Available: cds.caltech.edu/~nmatni/home/ACC16_LLQGX.pdf