Optimal Decentralized State-Feedback Control with Sparsity and Delays

Andrew Lamperski Laurent Lessard

Automatica, vol. 58, pp. 143–151, Aug. 2015

Abstract

This work presents the solution to a class of decentralized linear quadratic state-feedback control problems, in which the plant and controller must satisfy the same combination of delay and sparsity constraints. Using a novel decomposition of the noise history, the control problem is split into independent subproblems that are solved using dynamic programming. The approach presented herein both unifies and generalizes many existing results.

I Introduction

While optimal decentralized controller synthesis is difficult in general [25, 27], much progress has been made toward identifying tractable subclasses of problems. Two closely related conditions, partial nestedness and quadratic invariance, guarantee respectively that the optimal solution for an LQG control problem is linear [4], and that optimal synthesis can be cast as a convex program [16, 19]. These results alone do not guarantee that the optimal controller can be efficiently computed since the associated optimization problems are large.

For linear quadratic problems, efficient convex optimization methods have been used to solve state-feedback [17] and output-feedback [3, 8, 18] cases. A drawback of purely computational approaches is that little insight is gained into the structure of optimal controllers. However, efficient, explicit solutions that provide a physical interpretation for the states of the controller have been found separately for the delay and sparsity cases.

Delay case: All controllers eventually measure the global state, but not necessarily simultaneously. Instances with a one-timestep delay between controllers were solved in the 1970s [6, 20, 28]. In the linear quadratic setting, the state-feedback problem with delays characterized by a graph is solved in [7].

Sparsity case: All state measurements are transmitted instantly, but not all controllers receive all measurements. Explicit solutions for a two-controller system were given in [23] and extended to a general class of quadratically invariant sparsity patterns in [21, 22].

This paper unifies the treatment of state feedback with sparsity constraints, [21, 22], and delay constraints, [7], by considering an information flow characterized by a directed graph. Each edge may be labeled with a 0 for instantaneous information transfer or with a 1 for a one-timestep delay. See below for an example of such a graph. The 0–1 convention is merely for ease of exposition; the case of general inter-node delays is discussed in Section III.

Example 1. Consider the network graph of Fig. 1.

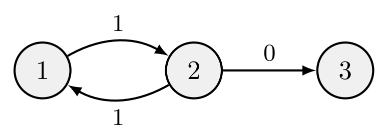

Figure 1: Network graph for Example 1. Each node represents a subsystem, and the edge labels indicate the propagation delay from one subsystem to another.

The example of Fig. 1 contains both salient features previously discussed: delay constraints (between nodes 1 and 2) and sparsity constraints (between nodes 2 and 3).

A fundamental assumption in this work is that the control policies are jointly optimized in order to minimize a global cost function. In our search for the optimal policies, we assume global knowledge of the graph topology, system dynamics, and cost function. In other words, the system is decentralized in the sense that controllers have limited state information at run time. However, the *design* of the controllers assumes global knowledge. In the absence of such an assumption the resulting problem is nonconvex [27]. Thus, work on multi-agent control with limited system knowledge typically does not study optimal control [1], or finds locally optimal solutions to nonconvex problems [2].

In Section II we sketch our approach for Example 1. In Sections III and IV, we treat general directed graphs. We discuss how our work unifies existing results in Section V and we discuss its limitations in Section VI. We prove the main results in Section VII and conclude in Section VIII. A preliminary version of this work appeared in [9]. The present work includes expanded proofs and discussions, and presents a message-passing implementation of the optimal controller.

II Solution to Example 1

The graph of Fig. 1 indicates constraints both on information sharing amongst controllers as well as on the system dynamics. In this case, the dynamics are given by discrete-time state-space equations of the form

$$\begin{bmatrix} x^1_{t+1} \\ x^2_{t+1} \\ x^3_{t+1} \end{bmatrix} = \begin{bmatrix} A^{11}_t & A^{12}_t & 0 \\ A^{21}_t & A^{22}_t & 0 \\ A^{31}_t & A^{32}_t & A^{33}_t \end{bmatrix} \begin{bmatrix} x^1_t \\ x^2_t \\ x^3_t \end{bmatrix} + \begin{bmatrix} B^{11}_t & B^{12}_t & 0 \\ B^{21}_t & B^{22}_t & 0 \\ B^{31}_t & B^{32}_t & B^{33}_t \end{bmatrix} \begin{bmatrix} u^1_t \\ u^2_t \\ u^3_t \end{bmatrix} + \begin{bmatrix} w^1_t \\ w^2_t \\ w^3_t \end{bmatrix} \quad (1)$$

for $t = 0, 1, \dots, T-1$. The state, input, and disturbance are denoted by x_t, u_t, and w_t respectively. Each vector is partitioned into subvectors associated with the nodes of the graph. For example, x^2_t is associated with node 2. The dynamics are constrained according to the directed graph. If node i cannot affect node j after a delay of 0 or 1, then $A^{ji}_t = 0$ and $B^{ji}_t = 0$ for all t.

We assume that for $i \in \{1, 2, 3\}$, the initial state and the disturbance vectors $\{x^i_0, w^i_0, \dots, w^i_{T-1}\}$ are independent Gaussian random vectors with means and covariances

$$x^i_0 \sim \mathcal{N}(0, \Sigma^i_0) \quad \text{and} \quad w^i_t \sim \mathcal{N}(0, W^i_t) \quad \text{for all } t. \quad (2)$$

The policies of the decision-makers choosing u^i_t are again constrained according to the graph. In particular,

$$u^1_t = \gamma^1_t\left(x^1_{0:t},\, x^2_{0:t-1}\right) \quad (3a)$$
$$u^2_t = \gamma^2_t\left(x^1_{0:t-1},\, x^2_{0:t}\right) \quad (3b)$$
$$u^3_t = \gamma^3_t\left(x^1_{0:t-1},\, x^2_{0:t},\, x^3_{0:t}\right) \quad (3c)$$

for all t, where each γ^i_t is a measurable function of the state information that has had sufficient time to propagate to node i. We use the notation $x^i_{0:t}$ to denote the state history $(x^i_0, \dots, x^i_t)$.

The objective is to choose the policies γ that minimize the expected finite-horizon quadratic cost

$$\min_\gamma \mathbb{E}^\gamma \left(\sum_{t=0}^{T-1} \begin{bmatrix} x_t \\ u_t \end{bmatrix}^\mathsf{T} \begin{bmatrix} Q_t & S_t \\ S_t^\mathsf{T} & R_t \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix} + x_T^\mathsf{T} Q_f x_T \right) \quad (4)$$

where the expectation is taken with respect to the joint probability measure on $(x_{0:T}, u_{0:T-1})$ induced by the choice of γ. We make the standard assumptions that

$$\begin{bmatrix} Q_t & S_t \\ S_t^\mathsf{T} & R_t \end{bmatrix} \geq 0, \qquad R_t > 0, \qquad Q_f \geq 0. \quad (5)$$

We assume that all decision-makers know the underlying network graph $G(\mathcal{V}, \mathcal{E})$ and all system parameters $A_{0:T-1}$, $B_{0:T-1}$, $Q_{0:T-1}$, $R_{0:T-1}$, $S_{0:T-1}$, and Q_f. Note that system matrix sizes may also vary with time.

Under the above assumptions, the problem is *partially nested*. Thus, the results from [4] imply that the optimal policies γ are linear functions.

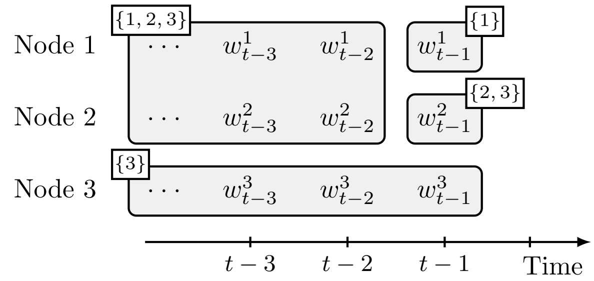

Figure 2: Noise partition diagram for Example 1 (see Fig. 1). The entire disturbance history is partitioned according to which subset of the nodes have access to the information. The subsets are indicated in the labels.

II-A Disturbance-feedback representation

The first step in our solution reparameterizes the input as functions the initial conditions and the disturbances. As in previous decentralized control work [3, 17, 21, 22], such a representation enables us to use statistical independence of the noise terms to simplify derivations.

Defining $w_{-1} := x_0$, the controllers (3) may be equivalently written as

$$u^1_t = \hat\gamma^1_t\left(w^1_{-1:t-1},\, w^2_{-1:t-2}\right) \quad (6a)$$
$$u^2_t = \hat\gamma^2_t\left(w^1_{-1:t-2},\, w^2_{-1:t-1}\right) \quad (6b)$$
$$u^3_t = \hat\gamma^3_t\left(w^1_{-1:t-2},\, w^2_{-1:t-1},\, w^3_{-1:t-1}\right) \quad (6c)$$

To see why, consider for example the information known by node 1 at time t. Given $(x^1_{0:t},\, x^2_{0:t-1})$, we may use (3) to compute past decisions $(u^1_{0:t-1}, u^2_{0:t-1})$. Then, using (1) we may infer the past disturbances $(w^1_{-1:t-1},\, w^2_{-1:t-2})$. Conversely, if $(w^1_{-1:t-1},\, w^2_{-1:t-2})$ is known, we may compute $(u^1_{0:t-1}, u^2_{0:t-1})$ via (6) and then compute $(x^1_{0:t},\, x^2_{0:t-1})$ via (3). It is straightforward to show that linearity of γ implies linearity of $\hat\gamma$.

II-B State and input decomposition

Extending the method from [7], we regroup the disturbance terms in order to decompose the input and state into independent random variables. Note that (6) can be used to partition the noise terms based on which subsets of the noise history the controllers can measure. This leads to the *noise partition diagram* shown in Fig. 2. For example, the bottom cluster $\{\dots, w^3_{t-3}, w^3_{t-2}, w^3_{t-1}\}$ is available only to u^3_t, whereas the cluster $\{w^2_{t-1}\}$ is available to both u^2_t and u^3_t. We call the noise subsets *label sets* and denote them by $\mathcal{L}^s_t$, where $s \in \{\{1\}, \{2,3\}, \{3\}, \{1,2,3\}\}$. For example, $\mathcal{L}^{\{1\}}_t = \{w^1_{t-1}\}$. We may rewrite (6) as

$$u^1_t = \hat\gamma^1_t\left(\mathcal{L}^{\{1\}}_t, \mathcal{L}^{\{1,2,3\}}_t\right) \quad (7a)$$
$$u^2_t = \hat\gamma^2_t\left(\mathcal{L}^{\{2,3\}}_t, \mathcal{L}^{\{1,2,3\}}_t\right) \quad (7b)$$
$$u^3_t = \hat\gamma^3_t\left(\mathcal{L}^{\{3\}}_t, \mathcal{L}^{\{2,3\}}_t, \mathcal{L}^{\{1,2,3\}}_t\right) \quad (7c)$$

Note that u_t^i depends on $\mathcal{L}_t^s$ if and only if $i \in s$. Because the disturbances are mutually independent and the label sets are disjoint, we may decompose u_t as a sum of its projections onto each of the $\mathcal{L}_t^s$. This leads to a decomposition of the form

$$u_t = \begin{bmatrix} \varphi_t^{\{1\}} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ [\varphi_t^{\{2,3\}}]^2 \\ [\varphi_t^{\{2,3\}}]^3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \varphi_t^{\{3\}} \end{bmatrix} + \varphi_t^{\{1,2,3\}} \tag{8}$$

where φ_t^s is a linear function of the elements of $\mathcal{L}_t^s$. Note that under this decomposition, the φ_t^s component of u_t^i is zero if $i \notin s$. We shall see that the states x_t^i also depend linearly on the label sets in a manner analogous to (7). Therefore, the state x_t can be similarly decomposed as

$$x_t = \begin{bmatrix} \zeta_t^{\{1\}} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ [\zeta_t^{\{2,3\}}]^2 \\ [\zeta_t^{\{2,3\}}]^3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \zeta_t^{\{3\}} \end{bmatrix} + \zeta_t^{\{1,2,3\}} \tag{9}$$

It will also be shown that the optimal decisions have the form $\varphi_t^s = K_t^s \zeta_t^s$ where the $\{K_t^s\}$ are real matrices and the equivalent constraints from (3), (6), and (7) are satisfied by construction.

II-C Update Equations

The optimality proof uses dynamic programming and requires a description of the evolution of ζ_t^s over time. Since ζ_t^s and φ_t^s are linear functions of the label set $\mathcal{L}_t^s$ terms, the dynamics of the label sets will be described as an intermediate step. From the noise partition diagram of Fig. 2, the label sets have dynamics

$$\begin{aligned} \mathcal{L}_{t+1}^{\{1\}} &= \{w_t^1\}, & \mathcal{L}_{t+1}^{\{3\}} &= \mathcal{L}_t^{\{3\}} \cup \{w_1^3\}, \\ \mathcal{L}_{t+1}^{\{2,3\}} &= \{w_t^2\}, & \mathcal{L}_{t+1}^{\{1,2,3\}} &= \mathcal{L}_t^{\{1,2,3\}} \cup \mathcal{L}_t^{\{1\}} \cup \mathcal{L}_t^{\{2,3\}} \end{aligned} \tag{10}$$

with initial conditions

$$\begin{aligned} \mathcal{L}_0^{\{1\}} &= \{x_0^1\}, & \mathcal{L}_0^{\{3\}} &= \{x_0^3\}, \\ \mathcal{L}_0^{\{2,3\}} &= \{x_0^2\}, & \mathcal{L}_0^{\{1,2,3\}} &= \emptyset. \end{aligned}$$

The label set dynamics can be visualized by using an *information graph* as shown in Fig. 3 (cf. [7]). An edge $r \to s$ indicates that $\mathcal{L}_t^r \subset \mathcal{L}_{t+1}^s$. Similarly, an edge $w^i \to s$ indicates that $\{w_t^i\} \subset \mathcal{L}_{t+1}^s$. It can be shown by induction that the ζ_t coordinates defined below satisfy (9) for $t = 0, \ldots, T$.

$$\zeta_{t+1}^{\{1\}} = w_t^1 \tag{11a}$$

$$\zeta_{t+1}^{\{2,3\}} = \begin{bmatrix} w_t^2 \\ 0 \end{bmatrix} \tag{11b}$$

$$\zeta_{t+1}^{\{3\}} = A_t^{33}\zeta_t^{\{3\}} + B_t^{33}\varphi_t^{\{3\}} + w_t^3 \tag{11c}$$

$$\begin{aligned} \zeta_{t+1}^{\{1,2,3\}} &= A_t\zeta_t^{\{1,2,3\}} + B_t\varphi_t^{\{1,2,3\}} \\ &\quad + A_t^{\{1,2,3\},\{2,3\}}\zeta_t^{\{2,3\}} + B_t^{\{1,2,3\},\{2,3\}}\varphi_t^{\{2,3\}} \\ &\quad + A_t^{\{1,2,3\},\{1\}}\zeta_t^{\{1\}} + B_t^{\{1,2,3\},\{1\}}\varphi_t^{\{1\}}, \end{aligned} \tag{11d}$$

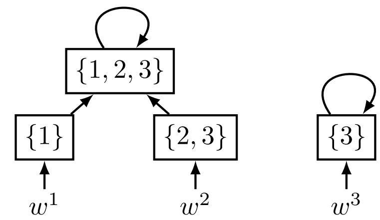

Figure 3: Information graph for Example 1. Each node corresponds to a subset of nodes in the network graph (see Fig. 1).

with initial conditions

$$\zeta_0^{\{1\}} = x_0^1, \; \zeta_0^{\{2,3\}} = \begin{bmatrix} x_0^2 \\ 0 \end{bmatrix}, \; \zeta_0^{\{3\}} = x_0^3, \; \zeta_0^{\{1,2,3\}} = 0.$$

The notation A_t^{rs} denotes the submatrix $[A_t^{ij}]$ with $i \in r$ and $j \in s$. For example, $A_t^{\{3\},\{2,3\}} = \begin{bmatrix} A_t^{32} & A_t^{33} \end{bmatrix}$.

Note that the dynamics in (11) can be deduced directly from the information graph. Indeed, ζ_{t+1}^s only depends on w_t^i if $w^i \to s$ and ζ_{t+1}^s only depends on (ζ_t^r, φ_t^r) whenever $r \to s$. If I is the identity matrix partitioned to conform with x_t, we have the following compact representation of the dynamics.

$$\zeta_0^s = \sum_{w^i \to s} I^{s,\{i\}} x_0^i \tag{12a}$$

$$\zeta_{t+1}^s = \sum_{r \to s} \left(A_t^{sr}\zeta_t^r + B_t^{sr}\varphi_t^r \right) + \sum_{w^i \to s} I^{s,\{i\}} w_t^i. \tag{12b}$$

II-D Decoupled Optimization Problems

Using the theory developed so far, we will sketch the strategy for decoupling optimization problems. The method is based on dynamic programming.

Suppose that the expected cost incurred by the optimal policy $\gamma_{0:T-1}^*$ for steps $t+1, \ldots, T$ has the form

$$\begin{aligned} &\mathbb{E}^{\gamma^*}\left(\sum_{k=t+1}^{T-1} \begin{bmatrix} x_k \\ u_k \end{bmatrix}^\mathsf{T} \begin{bmatrix} Q_k & S_k \\ S_k^\mathsf{T} & R_k \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + x_T^\mathsf{T} Q_f x_T \right) \\ &\qquad = \sum_s \mathbb{E}^{\gamma^*}\left(\left(\zeta_{t+1}^s\right)^\mathsf{T} X_{t+1}^s \zeta_{t+1}^s \right) + c_{t+1} \end{aligned} \tag{13}$$

where the X_{t+1}^s are positive semidefinite, c_{t+1} is a constant, and the sum ranges over the nodes of the information graph from Fig. 3. Using (9), this decomposition holds at $t+1 = T$ with $X_T^s = Q_f^{ss}$ and $c_T = 0$.

Substituting (12) into (13) and using independence, the expected cost for steps $t, t+1, \ldots, T$ is given by

$$\sum_r \mathbb{E}^{\gamma^*}\left(\begin{bmatrix} \zeta_t^r \\ \varphi_t^r \end{bmatrix}^\mathsf{T} \Gamma_t^r \begin{bmatrix} \zeta_t^r \\ \varphi_t^r \end{bmatrix} \right) + c_t, \tag{14}$$

where r ranges over all nodes in the information graph and Γ_t^r and c_t are given by

$$\Gamma_t^r = \begin{bmatrix} Q_t^{rr} & S_t^{rr} \\ S_t^{rr\mathsf{T}} & R_t^{rr} \end{bmatrix} + \begin{bmatrix} A_t^{sr} & B_t^{sr} \end{bmatrix}^{\mathsf{T}} X_{t+1}^s \begin{bmatrix} A_t^{sr} & B_t^{sr} \end{bmatrix} \tag{15}$$

$$c_t = c_{t+1} + \sum_{\substack{i \in \mathcal{V} \\ w^i \to s}} \textbf{trace}\left((X_{t+1}^s)^{\{i\},\{i\}} W_t^i \right). \tag{16}$$

Here s is the unique node such that $r \to s$.

Note that Γ_t^r is positive semidefinite, with a positive definite lower right block. It follows that the quadratic form in (14) is minimized over φ_t^r by a linear mapping

$$\varphi_t^r = K_t^r \zeta_t^r. \tag{17}$$

As discussed in Section II-B, the mapping (17) satisfies the information constraints of the problem and the optimal cost is of the form in (13).

II-E Message passing implementation

The controller described above depends on the ζ_t^r terms. These terms may be computed by using a combination of local measurements, local memory, and message passing. The proposed implementation may be visualized by augmenting Fig. 1 to include the appropriate messages, memory, and update equations. See Fig. 4.

In the rest of the paper, we will extend the results of this section more general decentralized control problems.

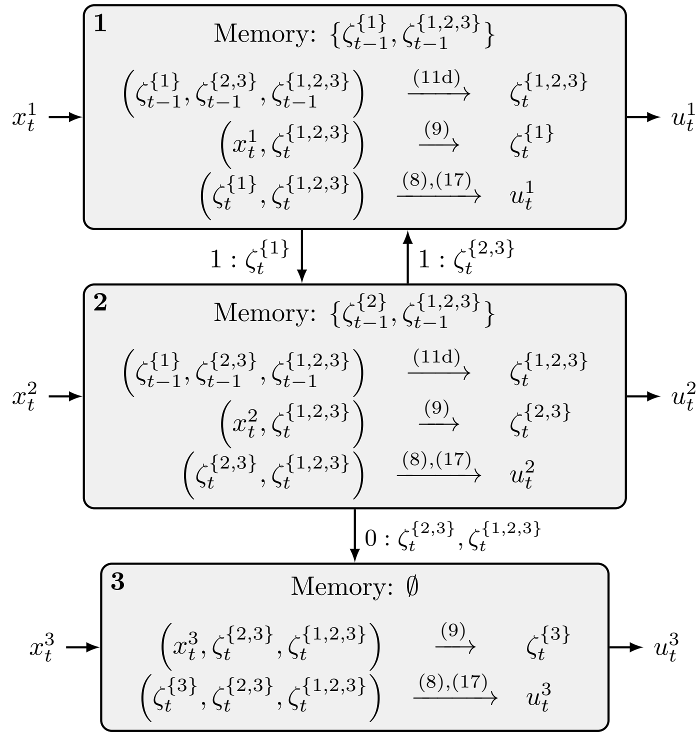

Figure 4: Network graph for Example 1 with messages and memory. The syntax $(\dots) \to \zeta_t^s$ means that ζ_t^s is computed as a function of the left-hand side terms. The numbers above the arrows indicate which equations are involved in the computation. The messages passed between nodes are shown next to the graph edges.

III Problem statement for the general case

We begin by defining some useful notation. The symbol I denotes a block-identity matrix whose dimensions are to be inferred by context. This notation is useful for extracting blocks from larger matrices. For example, if A_t is as in Example (1), the fact that $A_t^{13} = 0$ and $A_t^{23} = 0$ implies that $A_t I^{\{1,2,3\},\{3\}} = I^{\{1,2,3\},\{3\}} A_t^{33}$.

If $\mathcal{Y} = \{y^1, \dots, y^M\}$ is a set of random vectors (possibly of different sizes), we say that $z \in \textbf{lin}\,\mathcal{Y}$ if there are appropriately sized real matrices $C^1, \dots, C^M$ such that $z = C^1 y^1 + \cdots + C^M y^M$.

We also require some basic definitions regarding graphs. A *network graph* $G(\mathcal{V}, \mathcal{E})$ is a directed graph where each edge is labeled with a 0 if the associated link is delay-free, or a 1 if it has a one-timestep delay. The vertices are $\mathcal{V} = \{1, \dots, n\}$. If there is an edge from j to i, we write $(j, i) \in \mathcal{E}$, or simply $j \to i$. When delays are pertinent, they are denoted as $j \xrightarrow{0} i$ or $j \xrightarrow{1} i$. Directed cycles are permitted, but we assume there are no directed cycles with a total delay of zero. In our framework, all nodes in a delay-free cycle can be collapsed into a single node. Fig. 1 shows the network graph for Example 1. Associated with the network graph $G(\mathcal{V}, \mathcal{E})$ is the *delay matrix* D. Each entry D^{ij} is the sum of the delays along the directed path from j to i with the shortest delay. We assume $D^{ii} = 0$ for all i, and if no directed path exists, we set $D^{ij} = \infty$. The delay matrix for Example 1 is

$$D = \begin{bmatrix} 0 & 1 & \infty \\ 1 & 0 & \infty \\ 1 & 0 & 0 \end{bmatrix}. \tag{18}$$

Delays are assumed to be fixed for all time.

We now state the general class of problems that can be solved using the method developed in this paper.

Problem 1. Let $G(\mathcal{V}, \mathcal{E})$ be a network graph with associated delay matrix D. Suppose the following time-varying equations are given for all $i \in \mathcal{V}$ and for $t = 0, \dots, T-1$.

$$x_{t+1}^i = \sum_{\substack{j \in \mathcal{V} \\ D^{ij} \le 1}} \left(A_t^{ij} x_t^j + B_t^{ij} u_t^j \right) + w_t^i \tag{19}$$

Stacking the various vectors and matrices, we obtain the more compact representation

$$x_{t+1} = A_t x_t + B_t u_t + w_t. \tag{20}$$

The random vectors $\{x_0^i, w_0^i, \dots, w_{T-1}^i\}_{i \in \mathcal{V}}$ are mutually independent Gaussians, with means and covariances are given by (2). At time t controller i can only utilize state values from the *information set* defined by

$$\mathcal{I}_t^i = \{x_k^j : j \in \mathcal{V},\ 0 \le k \le t - D^{ij}\}, \tag{21}$$

so that for some function γ_t^i

$$u_t^i = \gamma_t^i(\mathcal{I}_t^i). \tag{22}$$

Note that $\mathcal{I}_t^i$ is the set of states belonging to nodes that have had sufficient time to reach node i by time t.

The goal is to choose the set of policies $\gamma = \{\gamma_{0:T-1}^i\}_{i\in\mathcal{V}}$ that minimize the expected quadratic cost (4)

In Problem 1, as in Example 1, we assume that all decision-makers know the system dynamics, cost matrices, and network graph.

While some feasible controllers could use memory that grows with the time horizon T, we will show by construction that there exists an optimal policy that has a finite memory that is independent of T.

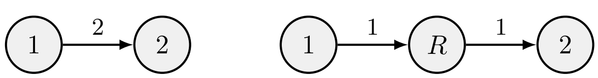

Figure 5: Network graph for a two-timestep delay. The equivalent representation on the right uses a relay node and two one-timestep delays.

Larger delays. The problem formulation in this paper only allows for delays of 0 or 1 timestep along edges of the network graph. Larger delays can be accommodated by including *relay* nodes. Consider for example the network graph of Fig. 5. At time t, the relay node stores x_{t-1}^1. Therefore, at time t, node 2 receives x_{t-2}^1 from the relay node, as desired. The system matrices for the relay node are also chosen so that there is no injected noise and no cost incurred.

IV Main results

This section presents the main results: two explicit state-space solutions for Problem 1. The first form has states which are functions of the primitive random variables x_0 and $w_{0:t-1}$. The second form uses state feedback only and gives a distributed implementation.

Our first step is to define the information graph (as in Fig. 3) for the general graph case (cf. [7]). Let s_k^j be the set of nodes reachable from node j within k steps:

$$s_k^j = \{i \in \mathcal{V} : D^{ij} \le k\}. \tag{23}$$

The information graph $\hat{G}(\mathcal{U}, \mathcal{F})$, is given by

$$\mathcal{U} = \{s_k^j : k \ge 0,\ j \in \mathcal{V}\}$$
$$\mathcal{F} = \{(s_k^j, s_{k+1}^j) : k \ge 0,\ j \in \mathcal{V}\}.$$

The additional labels w^i are not counted amongst the nodes of $\hat{G}$ as a matter of convention, but are shown as a reminder of which noise signal is being tracked. We will often write expressions such as $\{s \in \mathcal{U} : w^i \to s\}$ to denote the set of root nodes of $\hat{G}$. The following proposition gives some useful properties of the information graph.

Proposition 1. *Given an information graph $\hat{G}(\mathcal{U}, \mathcal{F})$, the following properties hold.*

(i) Every node in $\hat{G}$ has exactly one descendant. In other words, for every $r \in \mathcal{U}$, there is a unique $s \in \mathcal{U}$ such that $r \to s$.

(ii) Every path eventually hits a node with a self-loop.

(iii) If the network graph satisfies $|\mathcal{V}| = n$, the number of nodes in $\hat{G}$ is bounded by $n \le |\mathcal{U}| \le n^2 - n + 1$.

Note that the information graph will have several connected components whenever the network graph is not strongly connected, see Fig. 3.

We are now ready to present the main result of this paper, which expresses the optimal controller as a function of new coordinates induced by the information graph.

Theorem 2. *Consider Problem 1, and let $\hat{G}(\mathcal{U}, \mathcal{F})$ be the associated information graph. Define the matrices $\{X_{0:T}^r\}_{r\in\mathcal{U}}$ and $\{K_{0:T-1}^r\}_{r\in\mathcal{U}}$ recursively as follows,*

$$X_T^r = Q_f^{rr} \tag{24a}$$
$$\Omega_t^r = R_t^{rr} + B_t^{sr\mathsf{T}} X_{t+1}^s B_t^{sr} \tag{24b}$$
$$K_t^r = -\big(\Omega_t^r\big)^{-1}\big(S_t^{rr} + A_t^{sr\mathsf{T}} X_{t+1}^s B_t^{sr}\big)^\mathsf{T} \tag{24c}$$
$$X_t^r = Q_t^{rr} + A_t^{sr\mathsf{T}} X_{t+1}^s A_t^{sr} - K_t^{r\mathsf{T}} \Omega_t^r K_t^r \tag{24d}$$

where for each $r \in \mathcal{U}$, we have defined $s \in \mathcal{U}$ to be the unique node such that $r \to s$. The optimal control decisions satisfy the following state-space equations

$$\zeta_0^s = \sum_{w^i \to s} I^{s,\{i\}} x_0^i \tag{25a}$$
$$\zeta_{t+1}^s = \sum_{r \to s} \big(A_t^{sr} + B_t^{sr} K_t^r\big)\zeta_t^r + \sum_{w^i \to s} I^{s,\{i\}} w_t^i \tag{25b}$$
$$u_t^i = \sum_{r \ni i} I^{\{i\},r} K_t^r \zeta_t^r. \tag{25c}$$

The corresponding optimal expected cost is

$$V_0 = \sum_{\substack{i\in\mathcal{V} \\ w^i \to s}} \textbf{trace}\Big((X_0^s)^{\{i\},\{i\}} \Sigma_0^s\Big) + \sum_{t=0}^{T-1} \sum_{\substack{i\in\mathcal{V} \\ w^i \to s}} \textbf{trace}\Big((X_{t+1}^s)^{\{i\},\{i\}} W_t^i\Big). \tag{26}$$

Proof. See Section VII. ∎

Remark 3. *Note that when $r \in \mathcal{U}$ has a self-loop, the recursion for X_t^r only depends on X_{t+1}^r and is a classical Riccati equation. Otherwise, repeated application of (24) shows that X_t^r is a function of X_{t+k}^s, where $s \to s$ is the unique self loop reachable from r and k is the length of the path.*

Equation (25) expresses the controller as a map $w \mapsto u$. Our next result gives a message passing implementation of the optimal controller as a map $x \mapsto u$.

Theorem 4. *Consider the problem setting of Theorem 2. For each node i and all $t = 0, \dots, T-1$, define the outgoing message sent from node i to node j by*

$$\text{If } i \xrightarrow{0} j: \quad \mathcal{M}_t^{ij} = \{\zeta_t^s : s \in \mathcal{U},\, i,j \in s\}. \tag{27a}$$

$$\text{If } i \xrightarrow{1} j: \quad \mathcal{M}_t^{ij} = \{\zeta_t^s : s \in \mathcal{U},\, i \in s,\ j \notin s\}. \tag{27b}$$

and define the local memory of node i by $\mathcal{R}_0^i = \emptyset$ and

$$\mathcal{R}_{t+1}^i = \{\zeta_t^s : s \in \mathcal{U},\, i \in s,\, \nexists j \in s \text{ with } j \xrightarrow{0} i\}. \tag{28}$$

If controller i measures x_t^i at time t, then the distributed algorithm defined by (27) and (28) can be executed without deadlock. In other words, the $\mathcal{M}_t^{ij}$ and $\mathcal{R}_t^i$ can be computed for all t and i. Furthermore, if $i \in s \in \mathcal{U}$ then

$$\zeta_t^s \in \mathbf{lin}\left(\{x_t^i\} \cup \mathcal{R}_t^i \cup \bigcup_{j \xrightarrow{0} i} \mathcal{M}_t^{ji} \cup \bigcup_{j \xrightarrow{1} i} \mathcal{M}_{t-1}^{ji}\right), \tag{29}$$

where $\mathcal{M}_{-1}^{ji} = \emptyset$. Thus, the optimal u_t^i at every timestep can be computed from the measurement, the local memory, and the incoming messages at time t.

Proof. See Section VII-D. ∎ We will prove in Section VII that the optimal controller is unique. However, the choice of realization is not unique, and there is no guarantee that the representation given in Theorem 4 will be minimal.

The memory required by each node in Theorem 4 may be large because it depends on how many $s \in \mathcal{U}$ contain i. If the global state x_t has dimension N and there are n nodes, the memory is bounded by $|\mathcal{R}_t^i| \le n^2 N$. Note that this bound is independent of the horizon length T.

IV-A Extension to the infinite-horizon case

Our solution extends naturally to an infinite horizon when all system parameters, A, B, Q, R, S, and W are time-invariant. We seek a stabilizing controller that minimizes the average step cost as the horizon tends to ∞:

$$\min_{\gamma} \lim_{T\to\infty} \mathbb{E}^{\gamma}\left(\frac{1}{T}\sum_{t=0}^{T-1} \begin{bmatrix} x_t \\ u_t \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} Q & S \\ S^{\mathsf{T}} & R \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix}\right) \tag{30}$$

Corollary 5. *Consider Problem 1 under the time-invariance and average cost assumptions above and let $\hat{G}(\mathcal{U}, \mathcal{F})$ be the associated information graph. Further suppose that for each self-loop $s \to s$ in the information graph, the following assumptions hold:*

(1) (A^{ss}, B^{ss}) is stabilizable

(2) $\begin{bmatrix} A^{ss} - e^{\jmath\theta} I & B^{ss} \\ C^{ss} & D^{ss} \end{bmatrix}$ has full column rank $\forall \theta \in [0, 2\pi]$

where C^{ss} and D^{ss} are any matrices that factorize

$$\begin{bmatrix} Q^{ss} & S^{ss} \\ S^{ss\mathsf{T}} & R^{ss} \end{bmatrix} = \begin{bmatrix} C^{ss} & D^{ss} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} C^{ss} & D^{ss} \end{bmatrix}$$

Define the matrices $\{X^r\}_{r\in\mathcal{U}}$ and $\{K^r\}_{r\in\mathcal{U}}$ as follows

$$\Omega^r = R^{rr} + B^{sr\mathsf{T}} X^s B^{sr} \tag{31a}$$

$$K^r = -\big(\Omega^r\big)^{-1}\big(S^{rr} + A^{sr\mathsf{T}} X^s B^{sr}\big)^{\mathsf{T}} \tag{31b}$$

$$X^r = Q^{rr} + A^{sr\mathsf{T}} X^s A^{sr} - K^{r\mathsf{T}} \Omega^r K^r \tag{31c}$$

where for each $r \in \mathcal{U}$, we have defined $s \in \mathcal{U}$ to be the unique node such that $r \to s$. The optimal steady-state controller satisfies the following state-space equations

$$\zeta_{t+1}^s = \sum_{r\to s} \big(A^{sr} + B^{sr} K^r\big)\zeta_t^r + \sum_{w^i \to s} I^{s,\{i\}} w_t^i \tag{32a}$$

$$u_t^i = \sum_{r \ni i} I^{\{i\},r} K^r \zeta_t^r \tag{32b}$$

The corresponding optimal expected average cost is

$$V_0 = \sum_{\substack{i\in\mathcal{V} \\ w^i \to s}} \mathbf{trace}\Big(\big(X^s\big)^{\{i\},\{i\}} W^i\Big) \tag{33}$$

Proof. If $s \to s$ is a self-loop, then Remark 3 combined with the hypothesis implies that for any fixed t, as $T \to \infty$, the value of X_t^s converges to a stabilizing solution to the corresponding algebraic Riccati equation (31c), and $A^{ss} + B^{ss}K^s$ is stable, [29]. If r is not a self-loop, Remark 3 implies that X_t^r is a continuous function of X_{t+k}^s, and thus $X_t^r \to X^r$ as $T \to \infty$.

To see that the controller is stabilizing, note that when r is not a self-loop, then the mapping $w \mapsto \zeta^r$ has finite impulse response (FIR), and is thus stable. Thus if s is a self-loop, the mapping $w \to \zeta^s$ is of the form $\zeta_{t+1}^s = (A^{ss} + B^{ss}K^s)\zeta_t^s + \eta_t^s$, where $A^{ss} + B^{ss}K^s$ is stable and η_t^s FIR colored noise. ∎

V Specialization to existing results

In this section, we explain how Theorem 2 specializes to the existing results mentioned in Section I. Representative graphs for these examples are show in Fig. 6.

Corollary 6 (Centralized case)**.** *If the network graph has a single node as in Fig. 6a, the solution reduces to the standard linear quadratic regulator.*

Proof. The information graph consists of a single node with a self-loop. Thus, (24) reduces to the classical Riccati recursion and (25) implies that $\zeta_t^{\{1\}} = x_t^1$. ∎

Corollary 7 (Sparsity constraints [21, 22])**.** *If the network graph has N nodes with no delayed edges as in Fig. 6b, then the optimal gains can be computed from N classical Riccati recursions, one for each node.*

Proof. The information graph consists of N disconnected self-loops. Therefore, the solution from Theorem 2 reduces to N decoupled LQR solutions. ∎

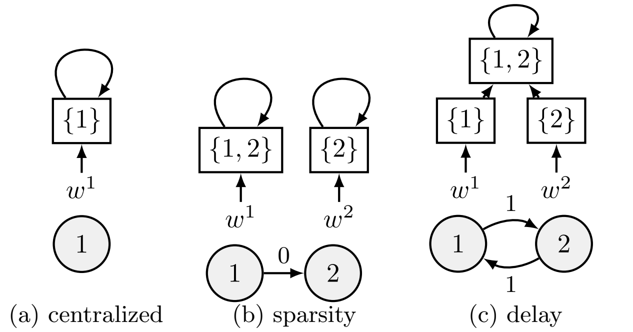

Figure 6: Three simple special cases.

Corollary 8 (Delay constraints [7])**.** *If the network graph is strongly connected and all edges have a one-timestep delay, then the optimal gains can be computed as algebraic functions of a single classical Riccati solution.*
Proof. All directed paths in the information graph lead to the self-loop $\mathcal{V} \to \mathcal{V}$. The recursions (24) imply that all gains can be computed as functions of $X_{0:T}^{\mathcal{V}}$, which is computed from a classical Riccati recursion. ∎

VI Limitations

We now discuss selected topics exploring the limitations of our work and directions for possible future research.

Output feedback. In output feedback problems, the decision-makers have access to noisy measurements of states rather than the states themselves. Solutions are known for several special topologies [5, 6, 10, 13, 14, 15, 20, 24, 28]. Despite these examples, it is unlikely that the present work will extend to output feedback for general graphs, since the decomposition of the information into independent sets is unlikely to hold.

Correlated noise. We assume in Problem 1 that the noises injected into the various nodes are independent. This fact implies that the ζ_t^s states are mutually independent, which simplifies the dynamic programming argument. If the noises are correlated, then the ζ_t^s may not be independent. Even for two player problems the optimal solutions have significantly different structures, [11].

Realizability. In general, a causal linear time-invariant system may be represented using either state-space or transfer functions. However, the two representations are not equivalent when we impose sparsity constraints for the state-space matrices [12, 26]. We avoid realizability issues by defining the problem in state space form, and derive a state space controller that satisfies the sparsity and delay constraints by construction.

VII Proof of main results

This section contains proofs of Theorems 2 and 4. The proof of Theorem 2 generalizes the sketch from Section II.

VII-A Linearity

Linearity of the optimal policy follows from *partial nestedness*, a concept first introduced by Ho and Chu in [4]. We state the main definition and result below.

Definition 9. *A dynamical system* (20) *with information structure* (22) *is* partially nested *if for every admissible policy γ, whenever u_τ^j affects $\mathcal{I}_t^i$, then $\mathcal{I}_\tau^j \subset \mathcal{I}_t^i$.*

Lemma 10 (see [4])**.** *Given a partially nested structure, the optimal control law that minimizes a quadratic cost of the form* (4) *exists, is unique, and is linear.*

In other words, an information structure is partially nested if whenever the decision of Player j affects the information used in Player i's decision, then Player i must have access to all the information available to Player j.

Lemma 11. *The information structure described in Problem 1 is partially nested, so the optimal solution is linear and unique.*

VII-B Disturbance-feedback representation

As in Section II-A, the control inputs as expressed as functions of the noise and initial conditions, in order to exploit independence properties.

Let $w_{-1}^i = x_0^i$, and define the *noise information set* by

$$\hat{\mathcal{I}}_t^i = \left\{ w_{k-1}^j : j \in \mathcal{V},\ 0 \le k \le t - D^{ij} \right\}. \tag{34}$$

Lemma 12. *A collection of functions $\{\gamma_{0:T-1}^i\}_{i\in\mathcal{V}}$ satisfies the information constraint* (22) *if and only if there are functions $\{\hat{\gamma}_{0:T-1}^i\}_{i\in\mathcal{V}}$ such that*

$$u_t^i = \gamma_t^i(\mathcal{I}_t^i) = \hat{\gamma}_t^i(\hat{\mathcal{I}}_t^i). \tag{35}$$

As in Section II, the paramterization in (35) is an intermediate step that will enable us to use a partition of the noise variables, defined in the next lemma, to decompose the inputs and states into independent variables.

Lemma 13. *Consider an information graph $\hat{G}(\mathcal{U}, \mathcal{F})$ and define the corresponding* label sets *$\{\mathcal{L}_{0:T}^s\}_{s\in\mathcal{U}}$ recursively by*

$$\mathcal{L}_0^s = \bigcup_{w^i \to s} \{x_0^i\} \tag{36a}$$

$$\mathcal{L}_{t+1}^s = \bigcup_{w_t^i \to s} \{w^i\} \cup \bigcup_{r \to s} \mathcal{L}_t^r. \tag{36b}$$

The following properties of the label sets hold.

(i) For every $t \ge 0$, the label sets are a partition of the noise history:

$$\mathcal{L}_t^r \cap \mathcal{L}_t^s = \emptyset \text{ when } r \ne s, \text{ and } \{w_{-1:t-1}\} = \bigcup_{s\in\mathcal{U}} \mathcal{L}_t^s.$$

(ii) For all $i \in \mathcal{V}$,

$$\hat{\mathcal{I}}_t^i = \bigcup_{s \ni i} \mathcal{L}_t^s. \tag{37}$$

Lemma 11 implies that the optimal solution is linear. When policies are restricted to be linear, (37) immediately implies the following corollary.

Corollary 14. *A linear policy $\{\gamma^i_{0:T-1}\}_{i\in\mathcal{V}}$ is feasible if and only if the inputs satisfy the following decomposition:*

$$u_t = \sum_{s\in\mathcal{U}} I^{\mathcal{V},s}\varphi^s_t, \tag{38}$$

where $\varphi^s_t \in \mathbf{lin}(\mathcal{L}^s_t)$.

As before, the state can also be decomposed as a sum of terms from $\mathbf{lin}(\mathcal{L}^s_t)$.

Lemma 15. *Say that $\varphi^s_t \in \mathbf{lin}(\mathcal{L}^s_t)$, and define ζ^s_t recursively by*

$$\zeta^s_0 = \sum_{w^i\to s} I^{s,\{i\}}x^i_0 \tag{39a}$$

$$\zeta^s_{t+1} = \sum_{r\to s}\left(A^{sr}_t\zeta^r_t + B^{sr}_t\varphi^r_t\right) + \sum_{w^i\to s} I^{s,\{i\}}w^i_t. \tag{39b}$$

Then $\zeta^s_t \in \mathbf{lin}(\mathcal{L}^s_t)$ and x_t can be decomposed as

$$x_t = \sum_{s\in\mathcal{U}} I^{\mathcal{V},s}\zeta^s_t. \tag{40}$$

Note that (39) agrees with the formula (25) from Theorem 2, provided that $\varphi^s_t = K^s_t\zeta^s_t$. Corollary 14 and Lemma 15 imply that this policy is feasible.

Remark 16. *We may interpret ζ^s_t and φ^s_t as conditional estimates of x_t and u_t, respectively. Namely,*

$$\zeta^s_t = I^{s,\mathcal{V}}\,\mathbb{E}(x_t \mid \mathcal{L}^s_t) \quad \textit{and} \quad \varphi^s_t = I^{s,\mathcal{V}}\,\mathbb{E}(u_t \mid \mathcal{L}^s_t).$$

VII-C Optimality

We now prove the controller is optimal, and derive an expression for the corresponding minimal expected cost. Our proof uses a dynamic programming argument, and we optimize over *policies* rather than *actions*. Let $\gamma_t = \{\gamma^i_t\}_{i\in\mathcal{V}}$ be the set of policies at time t. By Lemma 10, we may assume the γ^i_t are linear. Define the cost-to-go

$$V_t(\gamma_{0:t-1}) = \min_{\gamma_{t:T-1}} \mathbb{E}^\gamma\left(\sum_{k=t}^{T-1}\begin{bmatrix}x_k\\u_k\end{bmatrix}^\mathsf{T}\begin{bmatrix}Q_k & S_k\\ S_k^\mathsf{T} & R_k\end{bmatrix}\begin{bmatrix}x_k\\u_k\end{bmatrix} + x_T^\mathsf{T}Q_f x_T\right),$$

where the expectation is taken with respect to the joint probability measure on $(x_{t:T}, u_{t:T-1})$ induced by the choice of $\gamma = \gamma_{0:T-1}$. These functions are the minimum expected future cost from time t, given fixed policies up to time $t-1$. By iterating the minimizations we can write a recursive formulation for the cost-to-go,

$$V_t(\gamma_{0:t-1}) = \min_{\gamma_t}\mathbb{E}^\gamma\left(\begin{bmatrix}x_t\\u_t\end{bmatrix}^\mathsf{T}\begin{bmatrix}Q_t & S_t\\ S_t^\mathsf{T} & R_t\end{bmatrix}\begin{bmatrix}x_t\\u_t\end{bmatrix} + V_{t+1}(\gamma_{0:t-1},\gamma_t)\right). \tag{41}$$

Our objective is to find the optimal cost (4), which is simply V_0. Consider the terminal timestep, and use the decomposition (40),

$$V_T(\gamma_{0:T-1}) = \mathbb{E}^\gamma\left(x_T^\mathsf{T}Q_f x_T\right) = \mathbb{E}^\gamma\sum_{s\in\mathcal{U}}(\zeta^s_T)^\mathsf{T}Q^{ss}_f(\zeta^s_T).$$

In the last step, we used the fact that the ζ^s_t coordinates are independent. Note that V_T depends on the policies up to time $T-1$ because the distribution of ζ^s_T depends on past policies implicitly through (12b). We will prove by induction that the value function always has a similar quadratic form. Suppose that for some $t\geq 0$, we have

$$V_{t+1}(\gamma_{0:t}) = \mathbb{E}^\gamma\sum_{s\in\mathcal{U}}(\zeta^s_{t+1})^\mathsf{T}X^s_{t+1}(\zeta^s_{t+1}) + c_{t+1},$$

where $\{X^s_{t+1}\}_{s\in\mathcal{U}}$ is a set of matrices and c_{t+1} is a scalar. Now compute $V_t(\gamma_{0:t-1})$ using the recursion (41). Substituting φ^s_t and ζ^s_t from (38) and (38), and using the independence of $\mathcal{L}^s_t$, we obtain

$$V_t(\gamma_{0:t-1}) = \min_{\gamma_t}\mathbb{E}^\gamma\Bigg(\sum_{s\in\mathcal{U}}\begin{bmatrix}\zeta^s_t\\ \varphi^s_t\end{bmatrix}^\mathsf{T}\begin{bmatrix}Q^{ss}_t & S^{ss}_t\\ S^{ss\,\mathsf{T}}_t & R^{ss}_t\end{bmatrix}\begin{bmatrix}\zeta^s_t\\ \varphi^s_t\end{bmatrix} + (\zeta^s_{t+1})^\mathsf{T}X^s_{t+1}(\zeta^s_{t+1}) + c_{t+1}\Bigg).$$

Substituting the state equations (39b), using the independence and rearranging terms, we obtain

$$V_t(\gamma_{0:t-1}) = \min_{\gamma_t}\mathbb{E}^\gamma\sum_{r\in\mathcal{U}}\begin{bmatrix}\zeta^r_t\\ \varphi^r_t\end{bmatrix}^\mathsf{T}\Gamma^r_t\begin{bmatrix}\zeta^r_t\\ \varphi^r_t\end{bmatrix} + c_t, \tag{42}$$

where $\Gamma^r_{0:T-1}$ and $c_{0:T-1}$ are given by:

$$\Gamma^r_t = \begin{bmatrix}Q^{rr}_t & S^{rr}_t\\ S^{rr\,\mathsf{T}}_t & R^{rr}_t\end{bmatrix} + \begin{bmatrix}A^{sr}_t & B^{sr}_t\end{bmatrix}^\mathsf{T}X^s_{t+1}\begin{bmatrix}A^{sr}_t & B^{sr}_t\end{bmatrix} \tag{43}$$

$$c_t = c_{t+1} + \sum_{\substack{i\in\mathcal{V}\\ w^i\to s}}\mathbf{trace}\left((X^s_{t+1})^{\{i\},\{i\}}W^i_t\right). \tag{44}$$

The terminal conditions are $\Gamma^r_T = Q^{rr}_f$ and $c_T = 0$, and s is the unique node in $\hat{G}(\mathcal{U},\mathcal{F})$ such that $r\to s$, see Proposition 1. Note that the choice of K^r_t and X^r_t implies that the following bound holds pointwise:

$$\begin{bmatrix}\zeta^r_t\\ \varphi^r_t\end{bmatrix}^\mathsf{T}\Gamma^r_t\begin{bmatrix}\zeta^r_t\\ \varphi^r_t\end{bmatrix} \geq \begin{bmatrix}\zeta^r_t\\ K^r_t\zeta^r_t\end{bmatrix}^\mathsf{T}\Gamma^r_t\begin{bmatrix}\zeta^r_t\\ K^r_t\zeta^r_t\end{bmatrix} = (\zeta^r_t)^\mathsf{T}X^r_t(\zeta^r_t).$$

Substitution yields

$$V_t(\gamma_{0:t-1}) \geq \mathbb{E}^\gamma\sum_{s\in\mathcal{U}}(\zeta^s_t)^\mathsf{T}X^s_t(\zeta^s_t) + c_t.$$

This lower-bound is tight, because the optimal unconstrained actions are $\varphi^s_t = K^s_t\zeta^s_t \in \mathbf{lin}\,\mathcal{L}^s_t$, which is precisely the admissible set for φ^s_t. This completes the induction argument as well as the proof that the specified policy is optimal. The optimal cost is given by

$$V_0 = \mathbb{E}\sum_{s\in\mathcal{U}} (\zeta_0^s)^\mathsf{T} X_0^s (\zeta_0^s) + c_0$$
$$= \mathbb{E}\sum_{\substack{i\in\mathcal{V}\\ w^i\to s}} (x_0^i)^\mathsf{T} (X_0^s)^{\{i\},\{i\}} (x_0^i) + c_0. \tag{45}$$

where c_0 may be evaluated by starting with $c_T = 0$ and recursing backwards using (44). Finally, (45) evaluates to the desired expression (26) because $x_0^i \sim \mathcal{N}(0, \Sigma_0^i)$. This completes the proof of Theorem 2. ■

VII-D Proof of Theorem 4

Recall that there are no directed cycles in the network graph of delay 0. The proof will proceed by induction over the following partial order:

$$(t,i) \prec (\tau,j) \text{ if } \big(t<\tau\big) \text{ or } \big(t=\tau, i\neq j, \text{ and } D^{ji}=0\big).$$

Let $\tilde{\mathcal{I}}_t^i = \{x_t^i\} \cup \mathcal{R}_t^i \cup \bigcup_{j\xrightarrow{0}i} \mathcal{M}_t^{ji} \cup \bigcup_{j\xrightarrow{1}i} \mathcal{M}_{t-1}^{ji}$. If $t=0$ and i has no incoming delay-0 edges, then $\tilde{\mathcal{I}}_0^i = \mathcal{I}_0^i = \{x_0^i\}$. Thus (29), rewritten as $\zeta_t^s \in \mathbf{lin}(\tilde{\mathcal{I}}_t^i)$, holds at $(t,i) = (0,i)$.

Fix (t,i). Say that (29) holds for all (τ,j). with $(\tau,j) \prec (t,i)$. Agent i measures x_t^i directly, by assumption. If $j\xrightarrow{0}i$, then $(t,j)\prec(t,i)$ implies that $\mathcal{M}_t^{ji}$ could be computed and sent by agent j. If $t=0$, then the local memory and incoming delay-1 messages are empty. If $t>0$, then $(t-1,j)\prec(t,i)$ implies that the local memory $\mathcal{R}_t^i$ could be computed by agent i at time $t-1$ and that the messages $\mathcal{M}_{t-1}^{ji}$ could be computed as well. Thus, $\tilde{\mathcal{I}}_t^i$ can be computed.

Now it will be shown that (29) holds at (t,i). Say that $i\in s$. If $t=0$, then $\zeta_0^s \neq 0$ implies that ζ_0^s is a linear function of x_0^j, with $D^{ij}=0$. If $j=i$, then ζ_0^s can be computed from the local measurement, while if $j\neq i$, then ζ_0^s must have been contained in an incoming message.

Now say that $t>0$. First consider the case that $w^i \nrightarrow s$, so that (25b) reduces to

$$\zeta_t^s = \sum_{r\to s} \left(A_{t-1}^{sr} + B_{t-1}^{sr} K_{t-1}^r\right) \zeta_{t-1}^r.$$

If $i\notin r$, then ζ_{t-1}^r is contained in a delay-1 message $\mathcal{M}_{t-1}^{ji}$. So say that $i\in r$. If $\zeta_{t-1}^r \in \mathcal{R}_t^i$, then it is already available to agent i. Furthermore, if $\zeta_{t-1}^r \notin \mathcal{R}_t^i$, then it is contained in some message $\mathcal{M}_{t-1}^{ji}$, where $j\xrightarrow{0}i$. Since $i,j\in r\subset s$, it follows that ζ_t^s is contained in message $\mathcal{M}_t^{ji}$, so the equation above does not need to be computed. It follows that ζ_t^s may be computed from the combination of incoming messages and local memory.

Now consider the case that $w^i\to s$. The subvector $(\zeta_t^s)^{s\setminus\{i\}}$ can be computed as above using

$$(\zeta_t^s)^{s\setminus\{i\}} = I^{s\setminus\{i\},s} \sum_{r\to s}\left(A_{t-1}^{sr} + B_{t-1}^{sr}K_{t-1}^r\right)\zeta_{t-1}^r.$$

Since all vectors ζ_t^r with $i\in r\neq s$ can be computed as above, the subvector $(\zeta_t^s)^i$ can be computed using the state decomposition (40): $(\zeta_t^s)^i = x_i - \sum_{\substack{r\ni i\\ r\neq s}} (\zeta_t^r)^i$. Thus (29) holds at (t,i) and the proof is complete. ■

VIII Conclusion

This paper uses dynamic programming to derive optimal policies for a general class of decentralized linear quadratic state feedback problems. As noted in Section V, the solution generalizes many existing works on decentralized state-feedback control [7, 21, 22]. As discussed in Section VI, many possible avenues for future research remain open.

The key technique in the paper is the decomposition of available information based on the *information graph*. The graph is used to specify both dynamics of the controller states, as well as the structure of the Riccati difference equations required to compute the solution.

IX Acknowledgments

The first author thanks John Doyle and the second author thanks Ashutosh Nayyar for helpful discussions.

References

[1] Y. Cao, W. Yu, W. Ren, and G. Chen. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial Informatics*, 9(1), 2013.

[2] F. Farokhi, C. Langbort, and K. H. Johansson. Optimal structured static state-feedback control design with limited model information for fully-actuated systems. *Automatica*, 49:326–337, 2013.

[3] A. Gattami. Generalized linear quadratic control. *IEEE Transactions on Automatic Control*, 55(1):131–136, 2010.

[4] Y.-C. Ho and K.-C. Chu. Team decision theory and information structures in optimal control problems—Part I. *IEEE Transactions on Automatic Control*, 17(1):15–22, 1972.

[5] M. Kristalny and J. H. Cho. On the decentralized h^2 optimal control of bilateral teleoperation systems with time delays. In *IEEE Conference on Decision and Control*, pages 6908–6914, 2012.

[6] B.-Z. Kurtaran and R. Sivan. Linear-quadratic-Gaussian control with one-step-delay sharing pattern. *IEEE Transactions on Automatic Control*, 19(5):571–574, 1974.

[7] A. Lamperski and J. C. Doyle. Dynamic programming solutions for decentralized state-feedback LQG problems with communication delays. In *American Control Conference*, pages 6322–6327, 2012.

[8] A. Lamperski and J. C. Doyle. Output feedback $\mathcal{H}_2$ model matching for decentralized systems with delays. In *American Control Conference*, pages 5778–5783, 2013.

[9] A. Lamperski and L. Lessard. Optimal state-feedback control under sparsity and delay constraints. In *3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems*, pages 204–209, 2012.

[10] L. Lessard. Decentralized LQG control of systems with a broadcast architecture. In *IEEE Conference on Decision and Control*, volume 6241–6246, 2012.

[11] L. Lessard. Optimal control of a fully decentralized quadratic regulator. In *Allerton Conference on Communication, Control, and Computing*, pages 48–50, 2012.

[12] L. Lessard, M. Kristalny, and A. Rantzer. On structured realizability and stabilizability of linear systems. In *American Control Conference*, pages 5784–5790, 2013.

[13] L. Lessard and S. Lall. Optimal controller synthesis for the decentralized two-player problem with output feedback. In *American Control Conference*, pages 6314–6321, 2012.

[14] L. Lessard and S. Lall. Optimal control of two-player systems with output feedback. *ArXiv e-prints*, 2013. 1303.3644.

[15] L. Lessard and A. Nayyar. Structural results and explicit solution for two-player LQG systems on a finite time horizon. In *IEEE Conference on Decision and Control*, pages 6542 – 6549, 2013.

[16] X. Qi, M. V. Salapaka, P. G. Voulgaris, and M. Khammash. Structured optimal and robust control with multiple criteria: a convex solution. *IEEE Transactions on Automatic Control*, 49(10):1623–1640, 2004.

[17] A. Rantzer. Linear quadratic team theory revisited. In *American Control Conference*, pages 1637–1641, 2006.

[18] A. Rantzer. A separation principle for distributed control. In *IEEE Conference on Decision and Control*, pages 3609–3613, 2006.

[19] M. Rotkowitz and S. Lall. A characterization of convex problems in decentralized control. *IEEE Transactions on Automatic Control*, 51(2):274–286, 2006.

[20] N. R. Sandell and M. Athans. Solution of some nonclassical LQG stochastic decision problems. *IEEE Transactions on Automatic Control*, 19(2):108–116, 1974.

[21] P. Shah and P. A. Parrilo. $\mathcal{H}_2$-optimal decentralized control over posets: A state-space solution for state-feedback. *IEEE Transactions on Automatic Control*, 58(12):3084–3096, 2013.

[22] J. Swigart. *Optimal Controller Synthesis for Decentralized Systems*. PhD thesis, Stanford University, 2010.

[23] J. Swigart and S. Lall. An explicit state-space solution for a decentralized two-player optimal linear-quadratic regulator. In *American Control Conference*, pages 6385–6390, 2010.

[24] T. Tanaka and P. A. Parrilo. Optimal output feedback architecture for triangular LQG problems. In *American Control Conference*, pages 5730–5735, 2014.

[25] J. Tsitsiklis and M. Athans. On the complexity of decentralized decision making and detection problems. *IEEE Transactions on Automatic Control*, 30(5):440–446, 1985.

[26] A. S. M. Vamsi and N. Elia. Network realizability for interconnected systems over arbitrary one-step delay networks. In *American Control Conference*, pages 6252–6257, 2012.

[27] H. S. Witsenhausen. A counterexample in stochastic optimum control. *SIAM Journal on Control*, 6(1):131–147, 1968.

[28] T. Yoshikawa. Dynamic programming approach to decentralized stochastic control problems. *IEEE Transactions on Automatic Control*, 20(6):796–797, 1975.

[29] K. Zhou, J. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice-Hall, 1995.