Optimal State-Feedback Control Under Sparsity and Delay Constraints

Andrew Lamperski[1] Laurent Lessard[2]

3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems
(NecSys) pp. 204–209, 2012

Abstract

This paper presents the solution to a general decentralized state-feedback problem, in which the plant and controller must satisfy the same combination of delay constraints and sparsity constraints. The control problem is decomposed into independent subproblems, which are solved by dynamic programming. In special cases with only sparsity or only delay constraints, the controller reduces to existing solutions.

Notation

$y_{0:t}$	Time history of y: $\{y_0, y_1, \dots, y_t\}$.
$a \to b$	Directed edge from node a to node b.
d_{ij}	Shortest delay from node j to node i.
M^{sr}	Block submatrix $(M_{ij})_{i\in s, j\in r}$, e.g.

$$\begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix}^{\{2,3\},\{1\}} = \begin{bmatrix} M_{21} \\ M_{31} \end{bmatrix}.$$

1 Introduction

This paper studies a class of decentralized linear quadratic control problems. In decentralized control, inputs to a dynamic system are chosen by multiple controllers with access to different information. In this work, the dynamic system is decomposed into subsystems, each with a corresponding state and controller. A given controller has immediate access to some states, delayed access to some others, and no access to the rest. Since some controllers cannot access certain state components, the transfer matrix for the overall controller must satisfy sparsity constraints.

While decentralized synthesis is difficult in general, some classes of problems are known to be tractable and can be reduced to convex, albeit infinite dimensional, optimization problems. See [3] and [5]. These papers also suggest methods for solving the optimization problem. The former gives a sequence of approximate problems whose solutions converge to the global optimum, and the latter uses vectorization to convert the problem to a much larger one that is unconstrained. An efficient LMI method was also proposed by [4].

The class of problems studied in this paper have partially nested information constraints, which guarantees that linear optimal solutions exist, [1]. Explicit state-space solutions have been reported for certain special cases. For state-feedback problems in which the controller has sparsity constraints but is delay-free, solutions were given in [6], and [7]. For state-feedback with delays but no sparsity, a special case of the dynamic programming argument of this paper was given by [2].

In Sections 2 and 3, we state the problem and explain the state and input decomposition we will use. Main results appear in Section 4, followed by discussion and proofs.

2 Problem Statement

In this paper, we consider a network of discrete-time linear time-invariant dynamical systems. We will illustrate our notation and problem setup through a simple example, and then describe the problem setup in its full generality.

Example 1. Consider the state-space equations

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

For quantities with a time dependence, we use subscripts to specify the time index, while superscripts denote subsystems or sets of subsystems. Thus, x^i_t is the state of subsystem i at time t. The u^i_t are the inputs, and the w^i_t are independent Gaussian disturbances. The goal is to choose a state-feedback control policy that minimizes

[1]A. Lamperski is with the Department of Control and Dynamical Systems at the California Institute of Technology, Pasadena, CA 91125 USA. andyl@cds.caltech.edu

[2]L. Lessard is with the Department of Automatic Control at Lund University, Lund, Sweden. laurent.lessard@control.lth.se
The second author would like to acknowledge the support of the Swedish Research Council through the LCCC Linnaeus Center

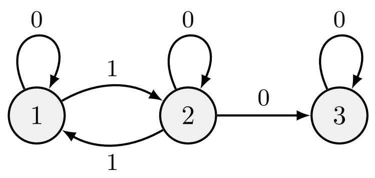


Figure 1: Network graph for Example 1. Each node represents a subsystem, and the arrows indicate the sparsity of both the dynamical interactions (1) as well as the information constraints (3). Additionally, the labels indicate the propagation delay from one controller to another.

the standard finite-horizon quadratic cost

$$\mathbb{E} \sum_{t=0}^{T-1} \begin{bmatrix} x_t \\ u_t \end{bmatrix}^\mathsf{T} \begin{bmatrix} Q & S \\ S^\mathsf{T} & R \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix} + x_T^\mathsf{T} Q_f x_T \tag{2}$$

with the usual requirement that R is positive definite, while $Q - SR^{-1}S^\mathsf{T}$ and Q_f are positive definite. Also, different controllers may have access to different information. For the purpose of this example, suppose the dependencies are as follows

$$\begin{aligned} u_t^1 &= \gamma_t^1\big(x_{0:t}^1,\, x_{0:t-1}^2\big) \\ u_t^2 &= \gamma_t^2\big(x_{0:t-1}^1,\, x_{0:t}^2\big) \\ u_t^3 &= \gamma_t^3\big(x_{0:t-1}^1,\, x_{0:t}^2,\, x_{0:t}^3\big) \end{aligned} \tag{3}$$

Note that there is a combination of *sparsity* and *delay* constraints; some state information may never be available to a particular controller, while other state information might be available but delayed. It is convenient to visualize this example using a directed graph with labeled edges. We call this the *network graph.* See Fig. 1.

Note that in Example 1, both the plant (1) and the controller (3) share the same sparsity constraints. Namely, x^3 does not influence x^1 or x^2 via the dynamics, nor can it affect u^1 or u^2 via the controller. This condition will be assumed for the most general case considered herein. As explained in Appendix A, this condition is sufficient to guarantee that the optimal control policy γ_t^i is linear, a very powerful fact.

A quantity of interest in this paper is the *delay matrix*, where each entry d_{ij} is defined as the sum of the delays along the fastest directed path from j to i. If no path exists, $d_{ij} = \infty$. In Example 1, the delay matrix is:

$$d = \begin{bmatrix} 0 & 1 & \infty \\ 1 & 0 & \infty \\ 1 & 0 & 0 \end{bmatrix} \tag{4}$$

This particular delay matrix only contains 0's, 1's, and ∞'s, but for more intricate graphs, the delay matrix can contain any nonnegative integer, as long as it is the total delay of the fasted directed path between two nodes.

General Case. In general, we consider any directed graph $G(V, E)$ with vertices $V = \{1, \ldots, N\}$. Every such edge is labeled with $d_{ij} \in \{0, 1\}$. We refer to this graph as the *network graph.* We then define d_{ij} for all other pairs of vertices according to the fastest-path definition above. We consider systems defined by the state-space equations:

$$x_{t+1}^i = \sum_{\substack{j \in V \\ d_{ij} \le 1}} \big(A_{ij} x_t^j + B_{ij} u_t^j\big) + w_t^i \quad \text{for all } i \in V \tag{5}$$

The initial state x_0 can be fixed and known, such as $x_0 = 0$, or it may be a Gaussian random variable. Note that the A_{ij} denote matrices, which can be viewed as submatrices of a larger A matrix. If we stack the various vectors and matrices, we obtain a more compact representation of the state-space equations:

$$x_{t+1} = Ax_t + Bu_t + w_t \tag{6}$$

where A and B have sparsity patterns determined by the d_{ij}. Namely, we can have $A_{ij} \neq 0$ whenever $d_{ij} \in \{0, 1\}$ and similarly for B_{ij}. This fact can be verified in Example 1 by comparing (1) and (4).

We assume the noise terms are Gaussian, IID for all time, and independent between subsystems. In other words, we have $\mathbb{E}(w_\tau^i w_t^{k\,\mathsf{T}}) = 0$ whenever $\tau \neq t$ or $i \neq k$, and $\mathbb{E}(w_t^i w_t^{i\,\mathsf{T}}) = W_i$ for all $t \geq 0$.

State information propagates amongst subsystems at every timestep according to the graph topology and the delays along the links. Each controller may use any state information that has had sufficient time to reach it. More formally, u_t^i is a function of the form

$$u_t^i = \gamma_t^i(x_{0:t-d_{i1}}^1, \ldots, x_{0:t-d_{iN}}^N). \tag{7}$$

Here if $t < d_{ij}$ (e.g. when $d_{ij} = \infty$), then u_t^i has no access to x^j at all. As a technical point, we assume that G contains no directed cycles with zero length. If it did, we could collapse the nodes of that cycle into a single node.

The objective is to find a control policy that satisfies the information constraints (7) and minimizes the quadratic cost function (2). According to our formulation, the controller may be any function of the past information history, which grows with the size of the time horizon T. However, we show in this paper how to construct an optimal controller that is linear and has a finite memory which does not depend on T.

3 Spatio-Temporal Decomposition

The solution presented herein depends on a special decomposition of the states and inputs into independent components. In Subsection 3.1, we define the *information graph*, which describes how disturbances injected at each

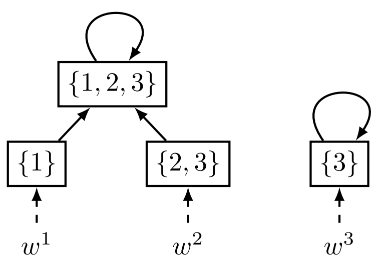
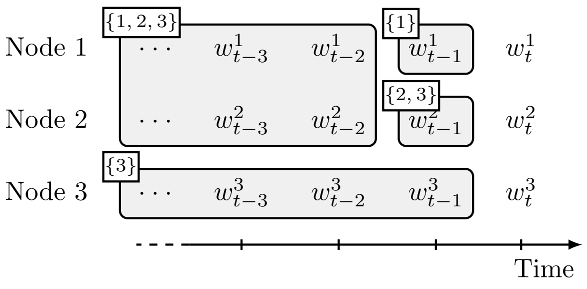


Figure 2: The information graph for Example 1. The nodes of this graph are the subsets of nodes in the network graph (see Fig. 1) affected by different noises. For example, w^2 injected at node 2 affects nodes $\{2,3\}$ immediately, and affects $\{1,2,3\}$ after one timestep.



Node 1
Node 2
Node 3
$\{1,2,3\}$
$\{1\}$
$\{2,3\}$
$\{3\}$
$\cdots$ w^1_{t-3} w^1_{t-2} w^1_{t-1} w^1_t
$\cdots$ w^2_{t-3} w^2_{t-2} w^2_{t-1} w^2_t
$\cdots$ w^3_{t-3} w^3_{t-2} w^3_{t-1} w^3_t
Time

Figure 3: Noise partition diagram for Example 1 (refer to Fig. 1 and 2). Each row represents a different node of the network graph, and each column represents a different time index, flowing from left to right. For example, w^2_{t-2} will reach all three nodes at time t, so it belongs to the label set $\{1,2,3\}$.

node propagate throughout the network. Then, in Subsection 3.2, we show how the information graph can be used to define a useful partition of the noise history. Finally, Subsection 3.3 explains the decomposition of states and inputs, which will prove crucial for our approach.

3.1 Information Graphs

The *information graph* a useful alternative way of representing the information flow in the network. Rather than thinking about nodes and their connections, we track the propagation of the noise signals w^i injected at the various nodes of the network graph. The information graph for Example 1 is shown in Fig. 2.

Formally, we define the information graph as follows. Let s^j_k be the set of nodes reachable from node j within k steps:

$$s^j_k = \{i \in V : d_{ij} \le k\}.$$

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

$$U = \{s^j_k : k \ge 0,\ \ j \in V\}$$
$$F = \{(s^j_k, s^j_{k+1}) : k \ge 0,\ \ j \in V\}.$$

We will often write $w^i \to s$ to indicate that $s = s^i_0$, though we do not count the w^i amongst the nodes of the information graph, as a matter of convention.

The information graph can be constructed by tracking each of the w^i as they propagate through the network graph. For example, consider w^2, the noise injected into subsystem 2. In Fig. 1, we see that the nodes $\{2,3\}$ are affected immediately. After one timestep, the noise has reached all three nodes. This leads to the path $w^2 \to \{2,3\} \to \{1,2,3\}$ in Fig. 2. When two paths reach the same subset, we merge them into a single node. For example, the path starting from w^1 also reaches $\{1,2,3\}$ after one step.

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

(i) Every node in the information graph has exactly one descendant. In other words, for every $r \in U$, there is a unique $s \in U$ such that $r \to s$.

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

(iii) If $d = \max_{ij}\{d_{ij} : d_{ij} < \infty\}$, then an upper bound for the number of nodes in the information graph is given by $|U| \le Nd$.

Example 2. Fig. 4 shows the network and information graphs for a more complex four-node network. We will refer to this example throughout the rest of the paper.

Note that the information graph may have several connected components. This happens whenever the network graph is not strongly connected. For example, Fig. 2 has two connected components because there is no path from node 3 to node 2.

3.2 Noise Partition

The *noise history* at time t is the set random variables consisting of all past noises:

$$H_t = \{w^i_\tau : i \in V,\ \ -1 \le \tau \le t-1\}$$

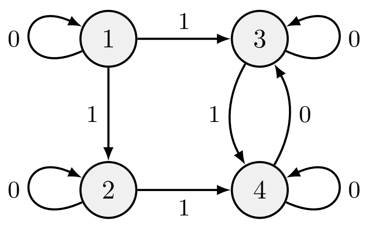
where we have used the convention $x_0 = w_{-1}$. We now define a special partition of H_t. This partition is related to the information graph; there is one subset corresponding to each $s \in U$. We call these subsets *label sets*, and they are defined in the following lemma.

Lemma 2. *For every $s \in U$, and for all $t \ge 0$, define the label sets recursively using*

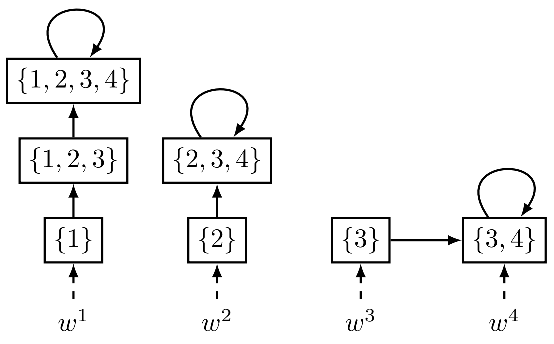
$$\mathcal{L}^s_0 = \{x^i_0 : w^i \to s\} \tag{8}$$
$$\mathcal{L}^s_{t+1} = \{w^i_t : w^i \to s\} \cup \bigcup_{r \to s} \mathcal{L}^r_t. \tag{9}$$

The label sets $\{\mathcal{L}^s_t\}_{s\in U}$ partition the noise history H_t:

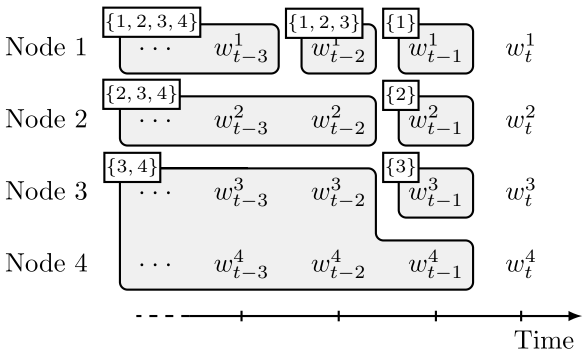
$$H_t = \bigcup_{s\in U} \mathcal{L}^s_t \qquad \text{and} \qquad \mathcal{L}^r_t \cap \mathcal{L}^s_t = \emptyset \quad \text{whenever } r \ne s$$



(a) Network graph for Example 2



(b) Information graph for Example 2



(c) Noise partition diagram for Example 2

Figure 4: Network graph (a), information graph (b), and noise partition diagram (c) for Example 2.

Proof. We proceed by induction. At $t = 0$, we have $H_0 = \{x_0^1, \dots, x_0^N\}$. For each $i \in V$, $s = s_0^i$ is the unique set such that $x_0^i \in \mathcal{L}_0^s$. Thus $\{\mathcal{L}_0^s\}_{s\in U}$ partitions H_0. Now suppose that $\{\mathcal{L}_t^s\}_{s\in U}$ partitions H_t for some $t \geq 0$. By Proposition 1, for all $r \in U$ there exists a unique $s \in U$ such that $r \to s$. Therefore each element $w_k^i \in H_t$ is contained in exactly one set $\mathcal{L}_{t+1}^s$. For each $i \in V$, we have that $\mathcal{L}_{t+1}^{s_0^i}$ is the unique label set containing w_t^i. Therefore $\{\mathcal{L}_{t+1}^s\}_{s\in U}$ must partition H_{t+1}. ∎

The partition defined in Lemma 2 can be visualized using a *noise partition diagram.* Example 1 is shown in Fig. 3 and Example 2 is shown in Fig. 4(c). These diagrams show the partition explicitly at time t by indicating which parts of the noise history belong to which label set. Each label set is tagged with its corresponding node in the information graph.

3.3 State and Input Decomposition

We show in Appendix A that our problem setup is *partially nested*, which implies that there exists an optimal control policy that is linear. We also show there that x_t and u_t are linear functions of the noise history H_t.

Individual components u_t^i will not depend on the full noise history H_t, because certain noise signals will not have had sufficient time to travel to node i. This fact can be read directly off the noise partition diagram. To see whether a noise symbol $w_{t-k}^i \in \mathcal{L}_t^s$ affects u_t^i, we simply check whether $i \in s$. We state this result as a lemma.

Lemma 3. *The input u_t^i depends on the elements of $\mathcal{L}_t^s$ if and only if $i \in s$. The state x_t^i depends on the elements of $\mathcal{L}_t^s$ if and only if $i \in s$.*

The noise partition described in Subsection 3.2 induces a decomposition of the input and state into components. We may write:

$$u_t = \sum_{s\in U} I^{V,s}\varphi_t^s \qquad \text{and} \qquad x_t = \sum_{s\in U} I^{V,s}\zeta_t^s \tag{10}$$

where φ_t^s and ζ_t^s each depend on the elements of $\mathcal{L}_t^s$. Here I is a large identity matrix with block rows and columns conforming to the dimensions of x_t^i or u_t^i depending on the context. The notation $I^{V,s}$ indicates the submatrix in which the block-rows corresponding to $i \in V$ and the block-columns corresponding to $j \in s$ have been selected. For example, the state in Example 1 can be written as:

$$x_t = \begin{bmatrix} I \\ 0 \\ 0 \end{bmatrix}\zeta_t^{\{1\}} + \begin{bmatrix} 0 & 0 \\ I & 0 \\ 0 & I \end{bmatrix}\zeta_t^{\{2,3\}} + \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix}\zeta_t^{\{3\}} + \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}\zeta_t^{\{1,2,3\}}$$

The vectors ζ_t^s each have different sizes, which is due to Lemma 3. For example, x_t^2 is only a function of $\zeta_t^{\{2,3\}}$ and $\zeta_t^{\{1,2,3\}}$, since these are the only label sets that contain 2.

We also use the superscript notation for other matrices. Taking (1) as an example, if $s = \{3\}$ and $r = \{2,3\}$, then $A^{sr} = \begin{bmatrix} A_{32} & A_{33} \end{bmatrix}$.

The state equations that define x_t (5) have a counterpart in the ζ_t coordinates. We state the following lemma without proof.

Lemma 4. *The components $\{\zeta_t^s\}_{s\in U}$ and $\{\varphi_t^s\}_{s\in U}$ satisfy the recursive equations:*

$$\zeta_0^s = \sum_{w^i\to s} I^{s,\{i\}} x_0^i \tag{11}$$

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

Two important properties of the input and state decomposition follow from results in this section. First, each input u_t^i is a function of a particular subset of the information history H_t, which follows directly from Lemma 3.

Corollary 5. *The input u_t^i depends on the elements of $\cup_{s \ni i} \mathcal{L}_t^s$. In particular, u_t^i has the ability to compute any state ζ_t^s for which $i \in s$.*

Secondly, Lemma 2 implies that the label sets for a given time index consist of mutually independent noises. So our decomposition provides independent coordinates:

Corollary 6. *Suppose $r, s \in U$, and $r \neq s$. Then:*

$$\mathbb{E} \begin{bmatrix} \zeta_t^r \\ \varphi_t^r \end{bmatrix} \begin{bmatrix} \zeta_t^s \\ \varphi_t^s \end{bmatrix}^\mathsf{T} = 0$$

4 Main Results

Consider the general problem setup described in Section 2, and generate the corresponding information graph $\hat{G}(U, F)$ as explained in Section 3.1. For every node $r \in U$, define the matrices $X_{0:T}^r$ recursively as follows:

$$\begin{aligned} X_T^r &= Q_f^{rr} \\ X_t^r &= Q^{rr} + A^{sr\mathsf{T}} X_{t+1}^s A^{sr} - \left(S^{rr} + A^{sr\mathsf{T}} X_{t+1}^s B^{sr}\right) \times \\ &\qquad \left(R^{rr} + B^{sr\mathsf{T}} X_{t+1}^s B^{sr}\right)^{-1} \left(S^{rr} + A^{sr\mathsf{T}} X_{t+1}^s B^{sr}\right)^\mathsf{T} \end{aligned} \tag{13}$$

where $s \in U$ is the unique node such that $r \to s$. Finally, define the gain matrices $K_{0:T-1}^r$ for every $r \in U$:

$$K_t^r = -\left(R^{rr} + B^{sr\mathsf{T}} X_{t+1}^s B^{sr}\right)^{-1} \left(S^{rr} + A^{sr\mathsf{T}} X_{t+1}^s B^{sr}\right)^\mathsf{T} \tag{14}$$

Theorem 7. *The optimal control policy is given by*

$$u_t = \sum_{s \in U} I^{V,s} \varphi_t^s \qquad \text{where} \qquad \varphi_t^s = K_t^s \zeta_t^s \tag{15}$$

where the states ζ_t^s evolve according to (12). The corresponding optimal cost is

$$V_0(x_0) = \sum_{\substack{i \in V \\ w_i \to s}} \left(\mathbf{trace}\Big((X_0^s)^{\{i\},\{i\}} \, \mathbb{E}\big(x_0^i x_0^{i\mathsf{T}}\big) \Big) + \sum_{t=1}^{T} \mathbf{trace}\Big((X_t^s)^{\{i\},\{i\}} W_i \Big) \right) \tag{16}$$

Theorem 7 describes an optimal controller as a function of the disturbances w. The controller can be transformed into an explicit state-feedback form as follows. For every $s \in U$, define the subset $s_w = \{i : w^i \to s\}$. This set will be a singleton $\{i\}$ when the node is a root node of the information graph, and will be empty otherwise. For example, consider Example 2 (Fig. 4(b)). When $s = \{3, 4\}$ we have $s_w = \{4\}$, and when $s = \{2, 3, 4\}$ we have $s_w = \emptyset$. Each $s \in U$ can be partitioned as $s = s_w \cup \overline{s_w}$, where $\overline{s_w} = s \setminus s_w$. The states of the controller are given by

$$\eta_t^{(s)} = I^{\overline{s_w},s} \zeta_t^s \qquad \text{for all } s \in U \text{ satisfying } \overline{s_w} \neq \emptyset$$

Theorem 8. *Let $\bar{A}_t^{sr} = A^{sr} + B^{sr} K_t^r$. The optimal state-feedback controller is given by the state-space equations*

$$\begin{aligned} \eta_0^{(s)} &= 0 \\ \eta_{t+1}^{(s)} &= \sum_{r \to s} I^{\overline{s_w},s} \bar{A}_t^{sr} I^{r,\overline{r_w}} \eta_t^{(r)} \\ &\quad + \sum_{r \to s} I^{\overline{s_w},s} \bar{A}_t^{sr} I^{r,r_w} \Bigg(x_t^{r_w} - \sum_{\substack{v \supset r \\ v \neq r}} I^{r_w,\overline{v_w}} \eta_t^{(v)} \Bigg) \end{aligned} \tag{17}$$

$$\begin{aligned} u_t &= \sum_{s \in U} I^{V,s} K_t^s I^{s,\overline{s_w}} \eta_t^{(s)} \\ &\quad + \sum_{s \in U} I^{V,s} K_t^s I^{s,s_w} \Bigg(x_t^{s_w} - \sum_{\substack{v \supset s \\ v \neq s}} I^{s_w,\overline{v_w}} \eta_t^{(v)} \Bigg) \end{aligned} \tag{18}$$

See Section 6 for complete proofs of Theorems 7 and 8.

5 Discussion

Note that for a self-loop, $r \to r$, (13) defines a classical Riccati difference equation. Since the information graph can have at most N self-loops, at most N Riccati equations must be solved. For other edges, (13) propagates the solution of the self-loop Riccati equation down the information graph toward the root nodes.

The solution extends naturally to infinite horizon, provided that at the self-loops, $r \to r$, the matrices satisfy classical conditions for stabilizing solutions to the corresponding algebraic Riccati equations. Indeed, as $T \to \infty$, X_t^{rr} approach steady state values. Since all gains are computed from the self-loop Riccati equations, the gains must also approach steady state limits.

The results can also be extended to time-varying systems simply by replacing A, B, Q, R, S, and W with A_t, B_t, Q_t, R_t, S_t, and W_t, respectively.

The results of [7] and [6] on control with sparsity constraints correspond to the special case of this work in which all edges have zero delay. In this case, the information graph consists of N self-loops. Thus, N Riccati equations must be solved, but they are not propagated.

The results on delayed systems of [2] correspond to the special case of strongly connected graphs in which all edges have a delay of one timestep. Here, all paths in the information graph eventually lead to the self-loop $V \to V$. Thus, a single Riccati equation is solved and propagated down the information graph.

6 Proofs of Main Results

6.1 Proof of Theorem 7

The proof uses dynamic programming and takes advantage of the decomposition described in Section 3. Begin by defining the cost-to-go from time t as a function of the current state and future inputs:

$$J_t(x_t, u_{t:T-1}) = \sum_{\tau=t}^{T-1} \begin{bmatrix} x_\tau \\ u_\tau \end{bmatrix}^\mathsf{T} \begin{bmatrix} Q & S \\ S^\mathsf{T} & R \end{bmatrix} \begin{bmatrix} x_\tau \\ u_\tau \end{bmatrix} + x_T^\mathsf{T} Q_f x_T$$

Note that J_t is a random variable because it does not depend explicitly on $x_{t+1:T}$. These future states are defined recursively using (6), and thus depend on the noise terms $w_{t:T-1}$. Using the decomposition 10, the state is divided according to $x_t = \sum_{s\in U} I^{V,s}\zeta_t^s$. We use the abridged notation ζ_t to denote $\{\zeta_t^s : s \in U\}$, and we use a similar notation to denote the decomposition of u_t into φ_t. In these new coordinates, the cost-to-go becomes:

$$J_t(\zeta_t, \varphi_{t:T-1}) = \sum_{s\in U} \left(\sum_{\tau=t}^{T-1} \begin{bmatrix} \zeta_\tau^s \\ \varphi_\tau^s \end{bmatrix}^\mathsf{T} \begin{bmatrix} Q^{ss} & S^{ss} \\ S^{ss\mathsf{T}} & R^{ss} \end{bmatrix} \begin{bmatrix} \zeta_\tau^s \\ \varphi_\tau^s \end{bmatrix} + \zeta_T^{s\,\mathsf{T}} Q_f^{ss} \zeta_T^s \right)$$

where the future states $\zeta_{t+1:T}$ are defined recursively using (12). Now define the value function, which is the minimum expected cost-to-go:

$$V_t(\zeta_t) = \min_{\varphi_{t:T-1}} \mathbb{E}\, J_t(\zeta_t, \varphi_{t:T-1})$$

Here, it is implied that the minimization is taken over *admissible* policies, namely φ_t^s is a linear function of $\mathcal{L}_t^s$ as explained in Section 3.3. Unlike the cost-to-go function, the value function is deterministic. The original minimum cost considered in the Introduction (2) is simply $V_0(x_0)$. The value function satisfies the Bellman equation

$$V_t(\zeta_t) = \min_{\varphi_t} \left(\mathbb{E} \sum_{s\in U} \begin{bmatrix} \zeta_t^s \\ \varphi_t^s \end{bmatrix}^\mathsf{T} \begin{bmatrix} Q^{ss} & S^{ss} \\ S^{ss\mathsf{T}} & R^{ss} \end{bmatrix} \begin{bmatrix} \zeta_t^s \\ \varphi_t^s \end{bmatrix} + V_{t+1}(\zeta_{t+1}) \right) \tag{19}$$

Lemma 9. *The Bellman equation* (19) *is satisfied by a quadratic value function of the form:*

$$V_t(\zeta_t) = \sum_{s\in U} \mathbb{E}\left(\zeta_t^{s\,\mathsf{T}} X_t^s \zeta_t^s \right) + c_t \tag{20}$$

Proof. The proof uses induction proceeding backwards in time. At $t = T$, the cost-to-go is simply the terminal cost $x_T^\mathsf{T} Q_f x_T$, and the value function is the expected value of this quantity. Decompose x_T into its ζ_T coordinates. The ζ_T coordinates are mutually independent by Corollary 6, so we may write

$$V_T(\zeta_T) = \sum_{s\in U} \mathbb{E}\left(\zeta_T^{s\,\mathsf{T}} Q_f^{ss} \zeta_T^s \right)$$

Thus, (20) holds for $t = T$, by setting $X_T^s = Q_f$ for every $s \in U$ and $c_T = 0$. Now suppose (20) holds for $t+1$. Equation (19) becomes:

$$V_t(\zeta_t) = \min_{\varphi_t} \left(\mathbb{E} \sum_{s\in U} \begin{bmatrix} \zeta_t^s \\ \varphi_t^s \end{bmatrix}^\mathsf{T} \begin{bmatrix} Q^{ss} & S^{ss} \\ S^{ss\mathsf{T}} & R^{ss} \end{bmatrix} \begin{bmatrix} \zeta_t^s \\ \varphi_t^s \end{bmatrix} + \mathbb{E} \sum_{s\in U} \zeta_{t+1}^{s\,\mathsf{T}} X_{t+1}^s \zeta_{t+1}^s + c_{t+1} \right) \tag{21}$$

Substitute the recursion for ζ_{t+1} defined in (12), and take advantage of the mutual independence of the ζ_t and φ_t terms proved in Corollary 6. Finally, we obtain:

$$V_t(\zeta_t) = \min_{\varphi_t} \left(\mathbb{E} \sum_{r\in U} \begin{bmatrix} \zeta_t^r \\ \varphi_t^r \end{bmatrix}^\mathsf{T} \Gamma_{t+1}^r \begin{bmatrix} \zeta_t^r \\ \varphi_t^r \end{bmatrix} \right) + c_{t+1} + \sum_{\substack{i\in V \\ w_i \to s}} \mathbf{trace}\left((X_{t+1}^s)^{\{i\},\{i\}} W_i \right) \tag{22}$$

where Γ_{t+1}^r is given by:

$$\Gamma_{t+1}^r = \begin{bmatrix} Q^{rr} & S^{rr} \\ S^{rr\mathsf{T}} & R^{rr} \end{bmatrix} + \begin{bmatrix} A^{sr} & B^{sr} \end{bmatrix}^\mathsf{T} X_{t+1}^s \begin{bmatrix} A^{sr} & B^{sr} \end{bmatrix}$$

and s is the unique node in the information graph such that $r \to s$, see Proposition 1. Equation (22) can be decomposed into independent quadratic optimization problems, one for each φ_t^r:

$$\varphi_t^r = \arg\min_{\varphi_t^r} \mathbb{E} \begin{bmatrix} \zeta_t^r \\ \varphi_t^r \end{bmatrix}^\mathsf{T} \Gamma_{t+1}^r \begin{bmatrix} \zeta_t^r \\ \varphi_t^r \end{bmatrix} \qquad \text{for all } r \in U$$

The optimal cost is again a quadratic function, which verifies our inductive hypothesis. ∎

The optimal inputs found by solving the quadratic optimization problems in Lemma 9 are given by $\varphi_t^s = K_t^s \zeta_t^s$. This policy is admissible by Corollary 5. Substituting the optimal policy, and comparing both sides of the equation, we obtain the desired recursion relation as well as the optimal cost (13)–(16). ∎

6.2 Proof of Theorem 8

Consider a node $s \in U$ with $w^i \to s$. Thus the set s_w is nonempty, and $s_w = \{i\}$. From (10),

$$x_t^i = \sum_{v \ni i} I^{s_w,v} \zeta_t^v = \sum_{v \supset s} I^{s_w,v} \zeta_t^v \tag{23}$$

The second equality follows because $v \ni i$ implies that $v \supset s$. Indeed, for any other $j \in s$, we must have $d_{ji} = 0$. Thus, if $v \ni i$, then $v \ni j$ as well. Rearranging (23) gives

$$I^{s_w,s} \zeta_t^s = x_t^i - \sum_{\substack{v\supset s \\ v \neq s}} I^{s_w,v} \zeta_t^v$$

Since s_w and $\overline{s_w}$ partition s, we may write ζ_t^s as

$$\begin{aligned}\zeta_t^s &= I^{s,\overline{s_w}} I^{\overline{s_w},s}\zeta_t^s + I^{s,s_w} I^{s_w,s}\zeta_t^s. \\ &= I^{s,\overline{s_w}} I^{\overline{s_w},s}\zeta_t^s + I^{s,s_w}\left(x_t^i - \sum_{\substack{v\supset s\\ v\neq s}} I^{s_w,v}\zeta_t^v\right) \\ &= I^{s,\overline{s_w}}\eta_t^{(s)} + I^{s,s_w}\left(x_t^i - \sum_{\substack{v\supset s\\ v\neq s}} I^{s_w,\overline{v_w}}\eta_t^{(v)}\right) \qquad (24)\end{aligned}$$

where we substituted the definition $\eta_t^{(s)} = I^{\overline{s_w},s}\zeta_t^s$ in the final step and used the fact that $i \in \overline{v_w}$, because $v \neq s$ and we can't have both $w^i \to v$ and $w^i \to s$.

We can now find state-space equations in terms of the new η_t coordinates. The input u_t is found by substituting (24) into (15), and the η_t dynamics are found by substituting (24) into (12) and multiplying on the left by $I^{\overline{s_w},s}$. Note that the controller now depends on the state x_t rather than the disturbance w_t. ■

References

[1] 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.

[2] Andrew Lamperski and John C. Doyle. Dynamic programming solutions for decentralized state-feedback LQG problems with communication delays. In *American Control Conference*, 2012.

[3] Xin 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.

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

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

[6] Parikshit Shah and Pablo A. Parrilo. $\mathcal{H}_2$-optimal decentralized control over posets: A state space solution for state-feedback. In *IEEE Conference on Decision and Control*, pages 6722–6727, 2010.

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

A Partial Nestedness

In this section, we discuss *partial nestedness*, a concept first introduced by [1]. Define the sets:

$$\mathcal{I}_t^i = \{x_\tau^j : j \in V,\ 0 \le \tau \le t - d_{ij}\}$$

Comparing with (7), $\mathcal{I}_t^i$ is precisely the information that u_t^i has access to when making its decision.

Definition 10. *A dynamical system* (6) *with information structure* (7) *is* partially nested *if for every admissible set of policies γ_t^i, whenever u_τ^j affects u_t^i, then $\mathcal{I}_\tau^j \subset \mathcal{I}_t^i$.*

The main result regarding partial nestedness is given in the following lemma.

Lemma 11 ([1]). *Given a partially nested structure, the optimal control for each member exists, is unique, and is linear.*

In order to apply this result, we must first show that the problems considered in this paper are of the correct type.

Lemma 12. *The information structure described in* (7) *is partially nested.*

Proof. Suppose u_τ^j affects u_t^i in the most direct manner possible. Namely, u_τ^j directly affects $x_{\tau+1}^\mu$, which affects a sequence of other states until it reaches x_σ^k, and $x_\sigma^k \in \mathcal{I}_t^i$. Further suppose that $x_\rho^\ell \in \mathcal{I}_\tau^j$.

$$\begin{aligned} u_\tau^j \text{ affects } x_\sigma^k \text{ directly} &\implies d_{kj} \le \sigma - \tau \qquad (25)\\ x_\sigma^k \in \mathcal{I}_t^i &\implies d_{ik} \le t - \sigma \qquad (26)\\ x_\rho^\ell \in \mathcal{I}_\tau^j &\implies d_{j\ell} \le \tau - \rho \qquad (27)\end{aligned}$$

Adding (25)–(27) together and using the triangle inequality, we obtain $d_{i\ell} \le t - \rho$. Thus, $x_\rho^\ell \in \mathcal{I}_t^i$. It follows that $\mathcal{I}_\tau^j \subset \mathcal{I}_t^i$, as required.

If u_τ^j affects u_t^i via a more complicated path, apply the above argument to each consecutive pair of inputs along the path to obtain the chain of inclusions $\mathcal{I}_\tau^j \subset \cdots \subset \mathcal{I}_t^i$. ■

With partial nestedness established, Lemma 11 implies that we have a unique linear optimal controller. In particular, the optimal γ_t^i are linear functions of $\mathcal{I}_t^i$. Looking back at (5), we conclude that x_t and u_t must be linear functions of the noise history $w_{-1:t-1}$, since w_t cannot affect x_t or u_t instantaneously. In general, individual components such as u_t^i will not be functions of the full noise history $w_{-1:t-1}$. This topic is discussed in detail in Section 3.