

Dynamic Programming Solutions for Decentralized State-Feedback LQG Problems with Communication Delays

Andrew Lamperski and John C. Doyle

Abstract—This paper presents explicit solutions for a class of decentralized LQG problems in which players communicate their states with delays. A method for decomposing the Bellman equation into a hierarchy of independent subproblems is introduced. Using this decomposition, all of the gains for the optimal controller are computed from the solution of a single algebraic Riccati equation.

I. INTRODUCTION

Decentralized control problems arise when control inputs to a dynamic system are chosen by multiple subsystems with access to different information. While decentralized control schemes are often difficult to synthesize, they arise in systems ranging from neural networks to the power grid.

In this paper, explicit optimal solutions for a class of decentralized control problems is found. Explicit solutions to decentralized optimal control problems are desirable because they can describe how components of large scale systems should behave if they are acting optimally. It is hoped that by discovering explicit solutions to a sufficiently rich set of examples, insight can be gained into control architectures arising in nature and engineering.

A. Related Work

This paper is a generalization of previous work by the authors which solved two simple LQG problems with communication delays by dynamic programming [1]. That paper as well as the current paper derive explicit solutions for a class of problems previously solved by semidefinite programming [2], [3]. While the existing solution is computationally efficient, the structure of the optimal controllers is not immediately clear from the semidefinite program. For some special cases of the work in this paper, the dynamic programming method can be extended to output feedback [4], [5], [6], but in general the extension is challenging because the separation principle often fails [7], [8]. The state feedback dynamic programming method presented in this paper is similar to methods developed for decentralized control with sparsity constraints [9].

B. Contributions

The main contribution of this paper is an explicit optimal controller for a general class of decentralized LQG control problems with communication delays. The input decomposes into a hierarchy of independent components which are defined by a graphical structure termed the information

hierarchy graph. Perhaps surprisingly, the only optimization required to compute the controller is the solution of a single discrete-time algebraic Riccati equation. The controller is found by propagating the solution of the Riccati equation through the information hierarchy graph.

C. Overview

The article is structured as follows. Section II defines the general problem studied in this paper. Section III defines information hierarchy graphs, which are used for decomposing information into independent components. Using the concept of information hierarchy graphs, the solution to problem of this paper is presented in Section IV. The solution is derived in Section V and finally conclusions are given in VI.

Notation. The expected value of a random variable, x , is denoted by $\mathbb{E}[x]$. The conditional expectation of x given y is denoted by $\mathbb{E}[x|y]$. Let $x(0:t)$ denote the stacked sequence of vectors: $x(0:t) = [x(0)^T \ x(1)^T \ \cdots \ x(t)^T]^T$.

For a vector partitioned into blocks, $[z_1^T \ \cdots \ z_n^T]^T$, and $v \subset \{1, \dots, n\}$, let $z^v = (z_i)_{i \in v}$. For instance, if $n = 5$ and $v = \{1, 3, 5\}$, then z^v is given by $z^{\{1,3,5\}} = [z_1^T \ z_3^T \ z_5^T]^T$.

For a matrix partitioned into blocks

$$M = \begin{bmatrix} M_{11} & \cdots & M_{1n} \\ \vdots & \ddots & \vdots \\ M_{n1} & \cdots & M_{nn} \end{bmatrix}$$

and $s, v \subset \{1, \dots, n\}$, let $M^{s,v} = (M_{i,j})_{i \in s, j \in v}$. For instance, if $n = 5$, $s = \{2, 4, 5\}$, and $v = \{3, 5\}$, then $M^{s,v}$ is given by

$$M^{\{2,4,5\},\{3,5\}} = \begin{bmatrix} M_{23} & M_{25} \\ M_{43} & M_{45} \\ M_{53} & M_{55} \end{bmatrix}.$$

II. PROBLEM STATEMENT

Consider a strongly connected directed graph $G = (V, E)$ with $|V| = n$, called a *delay structure graph*. Throughout this section, the graph in Figure 1 will be used as an example. It is assumed that one time-step is required for any piece of information to travel across an edge in the delay structure graph. Thus, if the shortest path from node i to node j has length d , then d time-steps are required for information to flow from node i to node j .

Associate a state vector $x_i \in \mathbb{R}^{k_i}$, an input vector $u_i \in \mathbb{R}^{p_i}$, and a process noise vector $w_i \in \mathbb{R}^{k_i}$ to each node in

A. Lamperski and J. C. Doyle are with Control and Dynamical Systems, California Institute of Technology, Pasadena, CA 91125 andyl@cds.caltech.edu

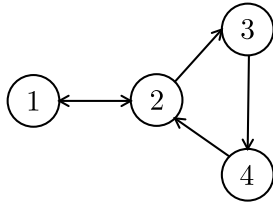


Fig. 1. A delay structure graph with four nodes. Each edge corresponds to a single step delay. So, one time-step is required for information to travel between nodes 1 and 2. Two time-steps are needed for information to travel from node 1 to node 3, and so on. Associated to each node i is a player which chooses an input u_i .

$i \in V$. The state vector is updated according to the following discrete-time dynamic equations:

$$x_i(t+1) = A_{ii}x_i(t) + \sum_{\{j:(j,i) \in E\}} A_{ij}x_j(t) + B_{ii}u_i(t) + w_i(t), \quad (1)$$

with initial conditions $x_i(0) = 0$. In Equation (1), A_{ij} and B_{ii} are matrices of appropriate dimension. Throughout the paper, $u_i(t)$ will be referred to as the input chosen by player i at time t .

For all $(i, j) \notin E$, let A_{ij} be the zero matrix of dimension $k_i \times k_j$. Then define the matrices A and B by

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{n1} & \cdots & A_{nn} \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & & & \\ & \ddots & & \\ & & \ddots & \\ & & & B_{nn} \end{bmatrix}.$$

By stacking x_i , u_i , and w_i into larger vectors,

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix}, \quad w = \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix},$$

Equation (1) can be written in the more compact form,

$$x(t+1) = Ax(t) + Bu(t) + w(t). \quad (2)$$

For the graph in Figure 1, A has the structure

$$A = \begin{bmatrix} A_{11} & A_{12} & 0 & 0 \\ A_{21} & A_{22} & 0 & A_{24} \\ 0 & A_{32} & A_{33} & 0 \\ 0 & 0 & A_{43} & A_{44} \end{bmatrix}.$$

To see how information flows around the graph based on the structure of A , consider a sparse vector, $w = [0 \ 0 \ * \ 0]^T$. The $*$ is used to indicate that the particular value of w is not important. It follows that successive applications of A give the following sparsity structures:

$$w = \begin{bmatrix} 0 \\ 0 \\ * \\ 0 \end{bmatrix}, \quad Aw = \begin{bmatrix} 0 \\ 0 \\ * \\ * \end{bmatrix}, \quad A^2w = \begin{bmatrix} 0 \\ * \\ * \\ * \end{bmatrix}, \quad A^3w = \begin{bmatrix} * \\ * \\ * \\ * \end{bmatrix}.$$

The process noise is Gaussian white noise, with terms corresponding to different nodes assumed to be uncorrelated:

$\mathbb{E}[w_i w_j^T] = 0$, when $i \neq j$. So the covariance of the noise, w , is given by

$$\mathbb{E}[ww^T] = \begin{bmatrix} W_1 & & & \\ & \ddots & & \\ & & & W_n \end{bmatrix}.$$

Let $d_{ii} = 0$, and let d_{ij} be the length of the shortest path from node i to node j . Since G is assumed to be strongly connected, d_{ij} must exist. The control problem is to minimize

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=0}^{N-1} \mathbb{E} [x(t)^T Q x(t) + u(t)^T R u(t)], \quad (3)$$

subject to the constraint that $\mathbb{E} [x(t)^T x(t)]$ remains bounded and inputs take the form

$$u_i(t) = \gamma_{i,t}(x_1(0:t-d_{1i}), \dots, x_n(0:t-d_{ni})). \quad (4)$$

Here, $\gamma_{i,t}$ are Borel-measurable functions to be chosen in the optimization procedure.

For the graph in Figure 1, the constraints on the input are given by

$$\begin{aligned} u_1(t) &= \gamma_{1,t}(x_1(0:t), x_2(0:t-1), \\ &\quad x_3(0:t-3), x_4(0:t-2)) \\ u_2(t) &= \gamma_{2,t}(x_1(0:t-1), x_2(0:t), \\ &\quad x_3(0:t-2), x_4(0:t-1)) \\ u_3(t) &= \gamma_{3,t}(x_1(0:t-2), x_2(0:t-1), \\ &\quad x_3(0:t), x_4(0:t-2)) \\ u_4(t) &= \gamma_{4,t}(x_1(0:t-3), x_2(0:t-2), \\ &\quad x_3(0:t-1), x_4(0:t)). \end{aligned}$$

The weight matrices Q and R are assumed to be partitioned into blocks, $Q = (Q_{ij})_{i,j \in V}$ and $R = (R_{ij})_{i,j \in V}$, conforming to the partitions of x and u , respectively. The matrix Q is positive semidefinite, and R is positive definite. To guarantee that a stabilizing solution to the corresponding algebraic Riccati equation exists, it will be assumed that (A, B) is stabilizable and (\sqrt{Q}, A) is detectable. No other assumptions are made about Q and R .

To derive the optimal controller, the following finite-horizon variant of the control problem is studied. Minimize

$$\mathbb{E} \left[\sum_{t=0}^{N-1} (x(t)^T Q x(t) + u(t)^T R u(t)) + x(N)^T \Lambda x(N) \right] \quad (5)$$

with inputs of the form of Equation (4). Here Λ is a positive semidefinite matrix of appropriate dimensions, corresponding to a terminal cost. Using a standard limiting argument, [10], it will be shown that as $N \rightarrow \infty$, the optimal controller for this finite-horizon problem approaches the steady-state controller.

Note that the assumptions about the structure of input and the sparsity structures of A and B guarantee that communication between the players choosing u_i occurs as least as fast as information travels through the plant. This

assumption implies that the information structure (the set of input constraints) is *partially nested*, which in turn implies that optimal inputs are linear in the associated information [11].

III. INFORMATION HIERARCHY GRAPHS

The optimal solution for the problem posed in Section II relies on an auxiliary structure, known as the information hierarchy graph, which can be derived from the original delay structure graph. This section presents the basic construction of the information hierarchy graph.

Let $G = (V, E)$ be the delay structure graph, with $V = \{1, \dots, n\}$. The *information hierarchy graph* $\mathcal{I} = (\mathcal{V}, \mathcal{E})$ is a graph describing the flow of information through G as constructed in Algorithm 1. See Figure 2 for a few examples of information hierarchy graphs constructed from their delay structure graphs.

Algorithm 1 Information Hierarchy Graph Construction

Start with $G = (V, E)$ and assume that $V = \{1, \dots, n\}$.
Set $\mathcal{V} = \{\{1\}, \dots, \{n\}\}$
Set $\mathcal{E} = \emptyset$
while There is a vertex $r \in \mathcal{V}$ with no outgoing edge **do**
 Pick $r \in \mathcal{V}$ with no outgoing edge
 Set $s = r$
 {Add to s all nodes reachable from nodes in v in one step}
 for all $i \in r$ **do**
 for all j such that $(i, j) \in E$ **do**
 if $j \notin s$ **then**
 Add j to s
 end if
 end for
 end for
 if $s \notin \mathcal{V}$ **then**
 Add s to \mathcal{V}
 end if
 Add edge (r, s) to \mathcal{E}
end while
return $\mathcal{I} = (\mathcal{V}, \mathcal{E})$

Remark 1: The information hierarchy graph can be used to describe which players have access to each piece of information. Let r be the set of nodes in V that are reachable from node i in k steps. It follows from Algorithm 1 that r is the unique node reachable from $\{i\}$ in \mathcal{I} in k steps. If a new piece of information becomes available to player i at time t , then it will be available to all players in the set r at time $t + k$.

Other useful properties information hierarchy graphs are now listed. All of the properties are direct consequences of Algorithm 1.

- Each node has exactly one outgoing edge.

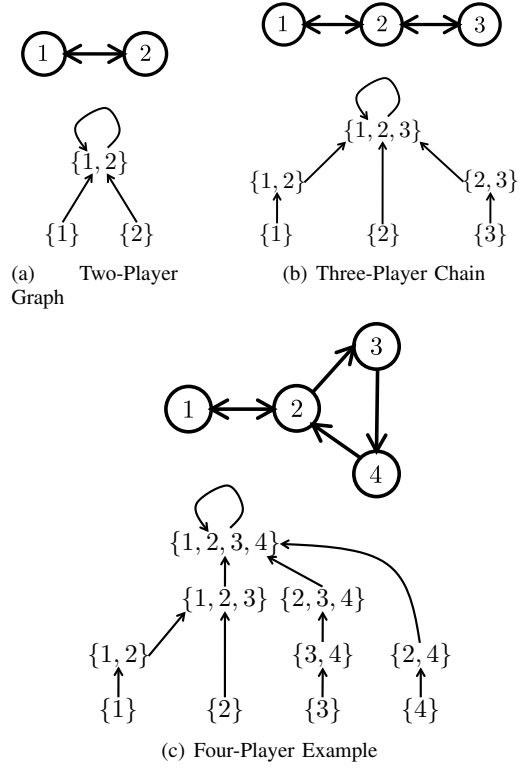


Fig. 2. Each subfigure depicts a delay structure graphs on the top with the associated information hierarchy graph on the bottom. Subfigures 2(a) and 2(b) correspond to the problems studied in [1].

- Nodes $\{1\}, \dots, \{n\}$ are the only nodes with no incoming edges.
- Since G is strongly connected, V is always a node in \mathcal{V} . Furthermore, the outgoing edge of V is a self-loop: $(V, V) \in \mathcal{E}$.
- $|\mathcal{V}| = |\mathcal{E}| \leq n(d + 1)$, where d is the longest path between any two nodes in E .

IV. OPTIMAL SOLUTION

This section presents the main result of the paper, Theorem 1, which gives the optimal controller for the problem defined by Equations (1), (3), and (4). The optimal solution is a dynamic controller that is constructed by propagating the solution to a standard Riccati equation through the information hierarchy graph.

Let X_V be the stabilizing solution to the discrete-time algebraic Riccati equation:

$$S = Q + A^T S A - A^T S B (R + B^T S B)^{-1} B^T S A. \quad (6)$$

Define the gain K_V by the standard LQR gain:

$$K_V = (R + B^T X_V B)^{-1} B^T X_V A. \quad (7)$$

For $r \neq V$, let s be the unique node such that $(r, s) \in \mathcal{E}$. Assume that X_s has already been defined and define X_r by

$$X_r = Q^{r,r} + A^{s,rT} X_s A^{s,r} - A^{s,rT} X_s B^{s,r} \left(R^{r,r} + B^{s,rT} X_s B^{s,r} \right)^{-1} B^{s,rT} X_s A^{s,r}. \quad (8)$$

Define the gain K_r by

$$K_r = \left(R^{r,r} + B^{s,rT} X_s B^{s,r} \right)^{-1} B^{s,rT} X_s A^{s,r}. \quad (9)$$

The gains, K_r , can now be used to define state equations for the optimal controller. Let $\zeta_s(t)$ be vectors, of the same dimension as $x^s(t)$, defined by the following dynamics:

$$\zeta_s(t+1) = \sum_{r:(r,s) \in \mathcal{E}} (A^{s,r} - B^{s,r} K_r) \zeta_r(t) \quad \text{for } s \in \mathcal{V} \text{ with } |s| > 1 \quad (10)$$

$$\zeta_{\{i\}}(t+1) = w_i(t) \quad \text{for } i = 1, \dots, n$$

with initial conditions $\zeta_s(0) = 0$.

Theorem 1: *The optimal controller for the general problem defined in Section II is given by*

$$u(t) = - \sum_{s \in \mathcal{V}} I_u^{V,s} K_s \zeta_s(t), \quad (11)$$

and the steady state cost is given by

$$\sum_{i=1}^n \text{Tr}(W_i X_{\{i\}}).$$

Here K_s and $X_{\{i\}}$ are defined by Equations (6)–(9), $\zeta_s(t)$ is defined by Equation (10), and I_u is the identity matrix partitioned into blocks conforming to the partition of $u(t)$.

V. CONTROLLER DERIVATION

This section derives the optimal controller presented in Section IV. First, in Subsection V-A, it is shown how to use the information hierarchy graph to decouple the information available to the players into independent components. Using this decomposition, the state and input are also decoupled into independent components. Next, in Subsection V-B, a finite-horizon version of the problem is solved via dynamic programming. Finally, in Subsection V-C, the steady state controller and optimal cost are derived by limiting arguments.

A. Decoupled State Dynamics

This subsection expands on the intuition from Remark 1 to describe a method for decoupling the information available to the players based on the information hierarchy graph. Once the information has been decoupled, the state and inputs are decomposed into independent terms. Finally, the dynamic equations for updating the decoupled state terms are given. The decoupled state variables will form the state of the controller.

For partially nested information structures, each player's optimal control is a linear function of the noise that influences that player's measurement [11]. Algorithm 2 shows how to label each node $s \in \mathcal{V}$ with a noise vector $\mathcal{L}_s(t)$ that can be computed by all players $i \in s$ but unavailable to all players $j \notin s$ (Figure 3). Note that the labels are pairwise independent, by construction. Lemmas 1 and 2 demonstrate that the labeling can be used to decompose the input into independent components.

Algorithm 2 Information Hierarchy Graph Labeling

Label nodes $\{1\}, \dots, \{n\}$ with $\mathcal{L}_{\{1\}}(t) = w_1(t-1), \dots, \mathcal{L}_{\{n\}}(t) = w_n(t-1)$, respectively.

while There is a node $s \in \mathcal{V} \setminus \{V\}$ that has not been labeled **do**

 Pick $s \in \mathcal{V} \setminus \{V\}$ such that s is not labeled and r is labeled for all r with $(r, s) \in \mathcal{E}$

for all r such that $(r, s) \in \mathcal{E}$ **do**

if The label for s has not been created **then**

 Set $\mathcal{L}_s(t) = \mathcal{L}_r(t-1)$

else

 Set $\mathcal{L}_s(t) = \begin{bmatrix} \mathcal{L}_s(t) \\ \mathcal{L}_r(t-1) \end{bmatrix}$

end if

end for

end while

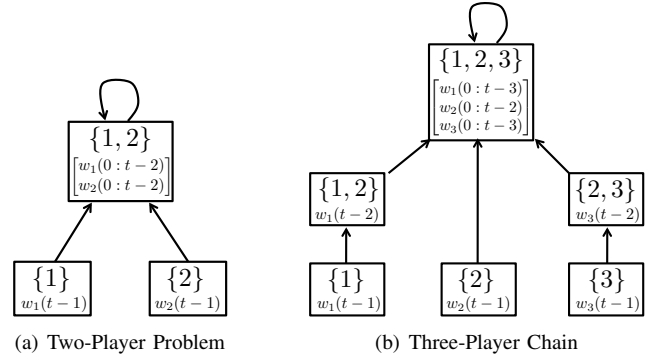
for $i = 1, \dots, n$ **do**

 Find s and k such that $(s, V) \in \mathcal{E}$ and $w_i(t-k)$ appears in $\mathcal{L}_s(t)$ $\{s$ and k will be unique}

 Set $d_i = k$

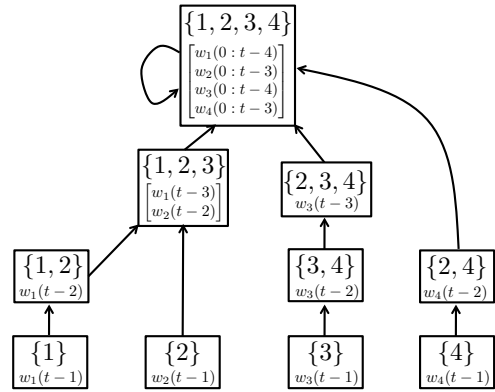
end for

Set $\mathcal{L}_V(t) = \begin{bmatrix} w_1(0:t-d_1-1) \\ \vdots \\ w_n(0:t-d_n-1) \end{bmatrix}$



(a) Two-Player Problem

(b) Three-Player Chain



(c) Four-Player Example

Fig. 3. Labeled information hierarchy graphs from Figure 2. The labels are pairwise independent and correspond to information available to all players in the corresponding node, but none of the other players.

Lemma 1: *Player i 's available information, from Equation (4), can depend on $\mathcal{L}_s(t)$ only if $i \in s$. Furthermore, if $i \in s$, then player i can calculate $\mathcal{L}_s(t)$.*

Proof: [Sketch] First note that if $s \neq V$ and $w_j(t-p-1) \in \mathcal{L}_s(t)$, then there must be a path in \mathcal{S} from $\{j\}$ to s of length p . Therefore, s is the set of nodes reachable from j in at most p steps.

Say that $i \notin s$ and take $w_j(t-p-1) \in \mathcal{L}_s(t)$. Thus any path from j to i has more than p steps. Using the information constraint, Equation (4), it can be shown that $w_j(t-p-1)$ must be independent of $u_i(t)$.

By Equation (1) and the fact that player i knows $x_i(t-1)$ and $x_j(t-1)$ for $(j, i) \in E$, it can compute $w_i(t-1) = \mathcal{L}_{\{i\}}(t)$. Now consider $w_j(t-p-1) \in \mathcal{L}_s(t)$ with $i \in s$. Equation (4) implies that player i has access to all the information available to player j at time $t-p$. In particular, player i can compute $w_j(t-p-1)$. ■

Lemma 2: *The optimal input $u(t)$ can be decomposed as a sum*

$$u(t) = \sum_{s \in \mathcal{V}} I_u^{V,s} \varphi_s(t), \quad (12)$$

where $\varphi_s(t)$ is a linear function of $\mathcal{L}_s(t)$ of appropriate size.

Proof: [Sketch] By linearity of the optimal solution, there exist matrices $H_{i,s}(t)$ such that the optimal input is given by

$$u_i(t) = \sum_{s \in \mathcal{V}: i \in s} H_{i,s}(t) \mathcal{L}_s(t).$$

Define $\varphi_s(t)$ by $I_u^{\{i\},s} \varphi_s(t) = H_{i,s}(t) \mathcal{L}_s(t)$. ■

Now that the input has been decomposed into independent terms, the state $x(t)$ can be similarly decomposed. Let $\zeta_s(t)$ be vectors, of the same dimension as $x^s(t)$, defined by the following dynamics:

$$\begin{aligned} \zeta_s(t+1) &= \sum_{r:(r,s) \in \mathcal{E}} (A^{s,r} \zeta_r(t) + B^{s,r} \varphi_r(t)) \\ &\quad \text{for } s \in \mathcal{V} \text{ with } |s| > 1 \end{aligned} \quad (13)$$

$$\zeta_{\{i\}}(t+1) = w_i(t) \quad \text{for } i = 1, \dots, n$$

with initial conditions $\zeta_s(0) = 0$ for all $s \in \mathcal{V}$. Equation (13) is the open loop counterpart of Equation (10).

Lemma 3: *The state vector can be decomposed as a sum*

$$x(t) = \sum_{s \in \mathcal{V}} I_x^{V,s} \zeta_s(t), \quad (14)$$

where I_x is the identity partitioned into blocks conforming to the partition of x , and $\zeta_s(t)$ is defined by Equation (13). Furthermore, $\zeta_s(t)$ is a linear function of $\mathcal{L}_s(t)$.

Proof: [Sketch] The proof is by induction. By the initial conditions, $\zeta_s(0) = 0$ and $x(0) = 0$, Equation (14) holds and $\zeta_s(t)$ is a linear function of $\mathcal{L}_s(t)$ at $t = 0$.

Now, inductively assume that Equation (14) holds at time t and that $\zeta_s(t)$ is a linear function of $\mathcal{L}_s(t)$. Plugging

Equations (12) – (14) into the dynamic equations shows that $x(t+1)$ is updated as follows:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + w(t) \\ &= \sum_{r \in \mathcal{V}} (AI_x^{V,r} \zeta_r(t) + BI_u^{V,r} \varphi_r(t)) + \\ &\quad \sum_{i=1}^n I_x^{V,\{i\}} \zeta_{\{i\}}(t+1). \end{aligned} \quad (15)$$

It can be shown by block matrix manipulations and the sparsity structures of A and B that $AI_x^{V,r} = I_x^{V,s} A^{s,r}$ and $BI_u^{V,r} = I_x^{V,s} B^{s,r}$. Plugging these identities into Equation (15) and applying Equation (13) to update ζ_s shows that

$$\begin{aligned} x(t+1) &= \sum_{|s|>1} \sum_{r:(r,s) \in \mathcal{E}} I_x^{V,s} (A^{s,r} \zeta_r(t) + B^{s,r} \varphi_r(t)) \\ &\quad + \sum_{i=1}^n I_x^{V,\{i\}} \zeta_{\{i\}}(t+1) \\ &= \sum_{s \in \mathcal{V}} I_x^{V,s} \zeta_s(t+1). \end{aligned}$$

The proof that $\zeta_s(t+1)$ is a linear function of $\mathcal{L}_s(t+1)$ follows from Equation (13) and Algorithm 2. ■

B. Finite-Horizon Dynamic Programming

Denote the optimal expected cost-to-go function by $\mathbb{E}[J(\zeta, t)]$. Recalling the finite-horizon cost function and plugging in the state decomposition of Equation (14), $\mathbb{E}[J(\zeta, N)]$ is given by

$$\begin{aligned} \mathbb{E}[J(\zeta, N)] &= \mathbb{E}[x^T \Lambda x] \\ &= \mathbb{E} \left[\left(\sum_{s \in \mathcal{V}} I_x^{V,s} \zeta_s \right)^T \Lambda \left(\sum_{s \in \mathcal{V}} I_x^{V,s} \zeta_s \right) \right] \\ &= \sum_{s \in \mathcal{V}} \mathbb{E}[\zeta_s^T \Lambda^{s,s} \zeta_s]. \end{aligned}$$

The last equality follows from the pairwise independence of ζ_s .

Set $X_s(N) = \Lambda^{s,s}$ for all $s \in \mathcal{V}$ and define $J(\zeta, N)$ to be $J(\zeta, N) = \sum_{s \in \mathcal{V}} \zeta_s^T X_s(N) \zeta_s$. Inductively assume that for some $t+1 \leq N$, $J(\zeta, t+1)$ is defined by

$$J(\zeta, t+1) = \sum_{s \in \mathcal{V}} \zeta_s^T X_s(t+1) \zeta_s + \sum_{k=t+2}^N \sum_{i=1}^n \text{Tr}(W_i X_{\{i\}}(k)). \quad (16)$$

The optimal expected cost-to-go function at time t is computed by solving the Bellman equation:

$$\mathbb{E}[J(\zeta, t)] = \min_{\varphi} \mathbb{E}[x^T Q x + u^T R u + J(\zeta', t+1)], \quad (17)$$

where ζ'_s are the variables ζ_s , updated according to Equation (13).

Substituting the decompositions for x and u and applying independence shows that the first two terms on the right-hand side can be decoupled as

$$\mathbb{E}[x^T Q x + u^T R u] = \sum_{s \in \mathcal{V}} [\zeta_s^T Q^{s,s} \zeta_s + \varphi_s^T R^{s,s} \varphi_s]. \quad (18)$$

Combining Equations (13) and (16), and applying independence shows that $\mathbb{E}[J(\zeta', t+1)]$ can be expanded as

$$\begin{aligned} & \mathbb{E}[J(t, \zeta(t+1))] \\ &= \sum_{r \in \mathcal{V}} \mathbb{E} [(A^{s,r} \zeta_r + B^{s,r} \varphi_r)^T X_s(t+1)] \\ & \quad (A^{s,r} \zeta_r + B^{s,r} \varphi_r) + \sum_{k=t+1}^N \sum_{i=1}^n \text{Tr}(W_i X_{\{i\}}(k)), \end{aligned} \quad (19)$$

where $(r, s) \in \mathcal{E}$.

Combining Equations (18) and (19) shows that the right-hand side of the Bellman equation can be decomposed into a sum of independent terms, plus a constant term:

$$\begin{aligned} & \min_{\varphi} \mathbb{E} [x^T Q x + u^T R u + J(\zeta', t+1)] = \\ & \sum_{r \in \mathcal{V}} \min_{\varphi_r} \mathbb{E} [\zeta_r^T Q^{r,r} \zeta_r + \varphi_r^T R^{r,r} \varphi_r + \\ & \quad (A^{s,r} \zeta_r + B^{s,r} \varphi_r)^T X_s(t+1) (A^{s,r} \zeta_r + B^{s,r} \varphi_r)] \\ & \quad + \sum_{k=t+1}^N \sum_{i=1}^n \text{Tr}(W_i X_{\{i\}}(k)). \end{aligned}$$

Standard quadratic minimization arguments show that the optimal inputs are given by

$$\varphi_r = -K_r(t) \zeta_r$$

with gains $K_r(t)$ computed as

$$K_r(t) = \left(R^{r,r} + B^{s,rT} X_s(t+1) B^{s,r} \right)^{-1} B^{s,rT} X_s(t+1) A^{s,r}.$$

Plugging in the inputs $\varphi_r = -K_r(t) \zeta_r(t)$ shows that $J(\zeta, t)$ has the form

$$J(\zeta, t) = \sum_{r \in \mathcal{V}} \zeta_r^T X_r(t) \zeta_r + \sum_{k=t+1}^N \sum_{i=1}^n \text{Tr}(W_i X_{\{i\}}(k))$$

where the matrices $X_r(t)$ are computed as follows (denoting $X_s(t+1)$ by X'_s to save space):

$$\begin{aligned} X_r(t) &= Q^{r,r} + A^{s,rT} X'_s A^{s,r} \\ & \quad - A^{s,rT} X'_s B^{s,r} \left(R^{r,r} + B^{s,rT} X'_s B^{s,r} \right)^{-1} B^{s,rT} X'_s A^{s,r}. \end{aligned}$$

Since $\mathbb{E}[J(\zeta, t+1)]$ was the optimal expected cost-to-go at time $t+1$, it follows inductively that $\mathbb{E}[J(\zeta, t)]$ is the optimal expected cost-to-go at time t , and the form of $J(\zeta, t)$ is valid for all $t \leq N$. Finally, since $x(0) = 0$, the total cost is calculated to be $\sum_{t=1}^N \sum_{i=1}^n \text{Tr}(W_i X_{\{i\}}(t))$.

C. Steady State

Note that $X_V(t)$ is the solution to the centralized LQR Riccati equation. Stabilizability of (A, B) and detectability of (\sqrt{Q}, A) imply that as $N \rightarrow \infty$, $X_V(t) \rightarrow X_V = S$, the stabilizing solution of the algebraic Riccati equation [10]. Since all the other matrices, $K_V(t)$, $X_r(t)$, and $K_r(t)$, are specified by $X_V(t)$, they respectively converge to the matrices K_V , X_r , and K_r , as defined in Equations (7), (8),

and (9), as $X_V(t) \rightarrow X_V$. Thus, the optimal gains and Riccati solutions have been found.

The steady state cost is calculated by noting that

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N \sum_{i=1}^n \text{Tr}(W_i X_{\{i\}}(t)) = \sum_{i=1}^n \text{Tr}(W_i X_{\{i\}}).$$

VI. CONCLUSION

This paper gives explicit optimal controllers for a class of state-feedback LQG problems whose delay structures are specified by graphs. To derive the optimal solution, the inputs are decomposed hierarchically based on the sharing of information. The top-level inputs have access to global, but delayed, state information, while lower-level inputs depend on newer, but more localized, information.

Future work will involve extending this work to more general delay patterns. Also, it would be desirable to unify the results of this paper with the sparsity constrained problems studied in [9], [12], [13]. Extension of the dynamic programming method in this paper to output feedback problems seems unlikely, in general, due to failure of the separation principle [7], [8]. Spectral factorization approaches to the output feedback problem are currently under investigation.

VII. ACKNOWLEDGEMENTS

The authors would like to thank Richard Murray, Sanjay Lall, and Pietro Perona for helpful discussions regarding the direction of this work.

REFERENCES

- [1] A. Lamperski and J. C. Doyle, "On the structure of state-feedback LQG controllers for distributed systems with communication delays," in *Conference on Decision and Control*, 2011.
- [2] A. Rantzer, "Linear quadratic team theory revisited," in *American Control Conference*, 2006.
- [3] A. Gattami, "Generalized linear quadratic control theory," in *IEEE Conference on Decision and Control*, 2006.
- [4] N. R. Sandell and M. Athans, "Solution of some nonclassical lqg stochastic decision problems," *IEEE Transactions on Automatic Control*, vol. 19, no. 2, pp. 108–116, 1974.
- [5] B.-Z. Kurtaran and R. Sivan, "Linear-quadratic-gaussian control with one-step-delay sharing pattern," *IEEE Transactions on Automatic Control*, vol. 19, no. 5, pp. 571–574, 1974.
- [6] T. Yoshikawa, "Dynamic programming approach to decentralized stochastic control problems," *IEEE Transactions on Automatic Control*, vol. 20, no. 6, pp. 796–797, 1975.
- [7] P. Varaiya and J. Walrand, "On delayed sharing patterns," *IEEE Transactions on Automatic Control*, vol. 23, no. 3, pp. 443–445, 1978.
- [8] T. Yoshikawa and H. Kobayashi, "Separation of estimation and control for decentralized stochastic control systems," *Automatica*, vol. 14, pp. 623–628, 1978.
- [9] J. Swigart and S. Lall, "An explicit dynamic programming solution for a decentralized two-player optimal linear-quadratic regulator," in *Symposium on the Mathematical Theory of Networks*, 2010, pp. 1443–1447.
- [10] F. L. Lewis and V. L. Syrmos, *Optimal Control*, 2nd ed. John Wiley & Sons, 1995.
- [11] Y.-C. Ho and K.-C. Chu, "Team decision theory and information structures in optimal control problems—part i," *IEEE Transactions on Automatic Control*, vol. 17, no. 1, 1972.
- [12] J. Swigart and S. Lall, "An explicit state-space solution for a decentralized two-player optimal linear-quadratic regulator," in *American Control Conference*, 2010, pp. 6385–6390.
- [13] P. Shah and P. Parrilo, " \mathcal{H}_2 -optimal decentralized control over posets: A state space solution for state-feedback," in *IEEE Conference on Decision and Control*, 2010.