On Computing Quadratic Controls for Acyclic Networks of Heterogeneous Systems

Iman Shames, Michael Cantoni

Abstract— **In this paper we consider the problem of computing finite-horizon quadratic optimal controls for a network of interconnected heterogeneous linear systems. We assume that the network is isomorphic to an acyclic directed graph and propose a solution to the problem via dynamic programming. This yields a finite sequence of smaller problems to be solved. Each smaller problem is associated with a node (or collection of such) in the network graph. It involves only local information and information from neighbours. As such, the solution algorithm lends itself to a distributed implementation. We numerically compare the performance of the proposed solution with computing the finite-horizon optimal controls by solving the global problem without exploiting structure.**

I. Introduction

The areas of distributed control and optimisation have been the subject of renewed interest over the last few years, and much effort has been invested in solving numerous practical problems in the aforementioned areas through implementing systems that take advantage of ubiquitous computing and networking capabilities. Smart-grids [1], water distribution networks [2], [3], and supply chains [4] are just a few examples of such problems.

In this paper, we consider the problem of quadratic control in a network of heterogeneous coupled linear systems. For networks with coupling over a directed acyclic graph, we propose a way to decompose the problem of computing the optimal input sequences for each node over the finite horizon. The decomposition is such that the computation can be distributed across the network, without requiring each node to maintain a global model of the system. Indeed, only local model information is required at each node, specifically, model information at each nodes and the information about that of those nodes that are coupled with it. Information that passes in sequence between neighbours (i.e. individual nodes in a path graph) is also required. The exchange of information is not part of an iterative process. It completes in a finite number of steps equal to, at most, twice the number of sub-systems or neighbourhoods. See [5], [6] for a review of techniques for distributing the computation associated with quadratic control. We compare the computational effort required by our proposed method with solving the global optimisation problem directly via numerical examples.

As mentioned above, we limit our results in this paper to those networks that are isomorphic to acyclic directed graphs, and particularly focus on the networks with underlying rooted trees. We later extend it networks isomorphic to other types of acyclic graphs. Moreover, the contributions of this paper can be seen in the light of the recent works where the problem of controller design in spatially-varying interconnected systems are considered and it is desired to obtain computationally efficient methods to obtain such controllers, e.g. see [7]–[9] and references there-in.

The inspiration behind the network topology and the coupling that we initially consider in this paper comes from the problem of controlling irrigation networks. An irrigation channel is composed of a set of interconnected pools that are separated by flume gates that can locally impose flow. Off-takes points for the supply of water to farms or secondary channels are located at the downstream end of pools. Hence, the interconnection resembles a rooted tree. For more information one may refer to [2], [10]. Furthermore, the different classes of graphs we study can be used to model many other physical systems. For example, the distribution section of an electrical grid can be modelled as a tree or a tree-like graph [11]. Cactus graphs can be used to model the relationships among genomes [12], and the importance of tree graphs in designing stable multi-agent formations under quantised communication is established in [13]. The problems considered subsequently are relevant to the application of receding horizon control techniques to network systems of these kind.

The outline of this paper is as follows. After introducing the notation used in this paper, we state the required preliminaries and the problem formulation in Section II. Then we address the problem described in Section II for a class of acyclic graphs in Section III. We present some simulation results and comparisons in Section IV. Concluding remarks and future research directions come in Section V.

Notations: In this paper capital letters are used to denote linear operators, small latin letter and greek letters represent vectors and scalars, and capital calligraphic letters represent sets, except for the well-known set $\mathbb{R}$. The cardinality of a set $\mathcal{A}$ is denoted by $|\mathcal{A}|$. Indices indicate a variable that is local to a subsystem with the same index, and the superscript $\star$ is used to indicate the optimal value of a variable with respect to a cost function. For integer variables $i = a : b$ means the same as $i = a, a+1, \ldots, b$ for integers a and b. We write $A > 0$ ($A \geq 0$) if A is Hermitian positive (semi)definite, and we write $A^\top$ and A^{-1} meaning the (conjugate) transpose and the (pseudo)inverse of A, respectively. By $x(t)$ and $x(t_1 : t_2)$ we mean the value of variable x at time t and the concatenation of values of variable x at times $t = t_1 : t_2$, respectively.

The authors are with the Department of Electrical and Electronic Engineering, the University of Melbourne. This work is supported in part by the Australian Research Council (DP110103778), a Uom Early Career Researcher grant, and a McKenzie fellowship.

II. PRELIMINARIES AND PROBLEM FORMULATION

Consider n interconnected discrete-time systems in set $\mathcal{V} = \{1:n\}$ where each system $i \in \mathcal{V}$ is described by

$$x_i(t+1) = A_i x_i(t) + B_i u_i(t) + E_i v_i(t) \tag{1}$$

where $x_i(t) \in \mathbb{R}^p$, $u_i(t) \in \mathbb{R}^m$, $v_i(t) \in \mathbb{R}^s$, $A_i \in \mathbb{R}^{p\times p}$, $B_i \in \mathbb{R}^{p\times m}$, $E_i \in \mathbb{R}^{p\times s}$, for some positive integers p, m, s, $x_i(t_0) = \xi_i$, and

$$v_i(t) = \sum_{l\in\mathcal{N}_i^-} U_{il} u_l(t) + \sum_{l\in\mathcal{N}_i^-} X_{il} x_l(t) \tag{2}$$

where U_{il} and X_{il} are coupling matrices of appropriate dimensions and describe the interconnection between i and the systems in some given set $\mathcal{N}_i^-$ (the graphical interpretation of this is given below.) and we assume that for any interconnection between i and l only one of U_{il} or X_{il} is nonzero. Additionally, we define $\mathcal{N}_i^+ = \{l | i \in \mathcal{N}_l^-\}$. Let $\mathcal{G}$ be either a directed or undirected non-empty graph. A path is a non-empty graph $\mathcal{P} = (\mathcal{V}_P, \mathcal{E}_P) \subset \mathcal{G}$ of the form $\mathcal{V}_P = \{i\}_{i=1}^k$ and $\mathcal{E}_p = \{(j_i, j_{i+1})\}_{i=1}^{k-1}$, where $\{j_1, \cdots, j_k\}$ is a permutation of $\{1, \cdots, k\}$. The vertices $j_2, \cdots, j_{k-1}$ are the inner vertices of $\mathcal{P}$. Furthermore, every sequence of edges that form a closed path in $\mathcal{G}$ and do not visit the same node twice, except the start/end node, is called cycle. A graph without any cycle is in turn termed *acyclic*.

The network of systems described by (1) and (2) can be modelled as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where the vertices of the graph are the (sub)systems, an edge connects two vertices if they are coupled, i.e. $\mathcal{E} = \{(l,i) | l, i \in \mathcal{V}, l \in \mathcal{N}_i^-\}$, $\mathcal{N}_i^-$ is the set of in-neighbours of i and $\mathcal{N}_i^+$ is the set of out-neighbours of i. The standing assumption throughout this paper is that $\mathcal{G}$ is acyclic, and , initially, $|\mathcal{N}_i^-| = 1$. Later we will comment on the situations where each i might have more than one in-neighbour.

Moreover, given $\tau \geq 1$ for each system i for $t = t_0+1 : t_0+\tau$, we have

$$x_i(t_0+1:t_0+\tau) = \begin{bmatrix} G_i & H_i & L_i \end{bmatrix} \begin{bmatrix} u_i(t_0:t_0+\tau-1) \\ v_i(t_0:t_0+\tau-1) \\ \xi_i \end{bmatrix} \tag{3}$$

where $G_i = \Gamma_i(B_i)$, $H_i = \Gamma_i(E_i)$, $\Gamma_i(\Delta) = \begin{bmatrix} \Delta_i & 0 & \dots & 0 \\ A_i\Delta_i & \Delta_i & \dots & 0 \\ \vdots & \dots & \dots & \vdots \\ A_i^{\tau-1}\Delta_i & A_i^{\tau-2}\Delta_i & \dots & \Delta_i \end{bmatrix}$, and $L_i = \begin{bmatrix} A_i \\ \vdots \\ A_i^{\tau} \end{bmatrix}$. For ease of notation we use x_i, u_i, and v_i instead of $x_i(t_0+1 : t_0+\tau)$, $u_i(t_0 : t_0+\tau-1)$, and $v_i(t_0 : t_0+\tau-1)$, respectively. Thus, (3) becomes

$$x_i = \begin{bmatrix} G_i & H_i & L_i \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ \xi_i \end{bmatrix}. \tag{4}$$

The main problem we consider in this paper is given below.

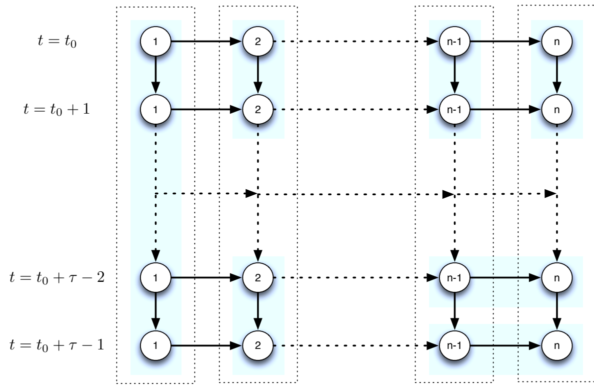

Fig. 1. The spatial and temporal interdependence of the systems in a path.

Problem 1: Consider the aforementioned interconnected systems. Given $\tau \geq 1$, define

$$\begin{aligned} f(u_1,\dots,u_n) &= \frac{1}{2}\sum_{i=1}^{n}\sum_{t=t_0+1}^{t_0+\tau} f_i(u_i(t)) \\ &= \frac{1}{2}\sum_{i=1}^{n}\sum_{t=t_0+1}^{t_0+\tau} x_i(t)^\top Q_i(t) x_i(t) + u_i(t)^\top R_i(t) u_i(t) \end{aligned} \tag{5}$$

for known $Q_i(t) > 0$ and $R_i(t) > 0$. Furthermore, let $v_0(t) = 0$, for all $t > 0$. Compute $u_i^\star$, $i = 1:n$, to minimise (5) subject to (2) and (4).

Moreover, we define Q_i and R_i to be block diagonal matrices with blocks $Q_i(t_0), \dots, Q_i(t_0+\tau-1)$ and $R_i(t_0), \dots, R_i(t_0+\tau-1)$, respectively.

III. DISTRIBUTED FINITE-HORIZON OPTIMAL CONTROL FOR CLASSES OF ACYCLIC DIRECTED GRAPHS

In this section we address Problem 1 for different classes of directed acyclic graphs. The first class of networks we consider are those that can be represented by a directed path.

A. Directed Paths

We initially consider the case where $\mathcal{N}_i^- = \{i-1\}$ and $\mathcal{N}_i^+ = \{i+1\}$. This assumption renders the above-mentioned graph to be an elementary path from i to n. Additionally, we write U_i and X_i meaning $U_{i,i-1}$ and $X_{i,i-1}$ respectively. We denote this path by $\mathcal{P} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{i\}_{i=1}^n$ and $\mathcal{E} = \{(i, i+1) | i, i+1 \in \mathcal{V}\}$. We also consider different coupling scenarios via different configurations of (2). However, before continuing further, we comment on the way that the principles of dynamic programming are employed in this paper. Fig. 1 depicts the way that systems in the aforementioned path are interconnected spatially and temporally for time $t = t_0 : t_0+\tau-1$. Each system i at time t directly influences itself at time $t+1$ and system $i+1$ at time t. That is, there is both temporal and spatial causality. The more conventional way of calculating the optimal inputs would be to characterise the optimal inputs of the subsystems at temporal stage $t = t_0+\tau-1$ and apply the dynamic programming principle to characterise the optimal inputs at each preceding temporal stage. Proceeding in this way would yield a solution algorithm, which would not lend itself to distributed implementation, in that each node would need access to global network model information and at each stage of the computation all-to-all communication would be required. Thus, instead of this temporal treatment of the interconnected systems, one can apply the dynamic programming principle spatially. Such an application of the dynamic programming enables each systems i to calculate its optimal input via its locally available information and information that is acquired from neighbours only. Moreover, even if the ability to efficiently perform calculations in a distributed fashion would not be of importance, for the situations where $\tau \ll n$, such a spatial view result in much smaller problems, and hence more efficient algorithms, than the more conventional temporal approach.

1) Coupling Through Control Signals: In this sub-section we consider the scenarios that the intersystem coupling is through the control inputs of the neighbouring systems. First we consider the case where $v_i(t) = U_i u_{i-1}(t)$. Without loss of generality one can assume U_i to be the identity matrix as for the case where it is not equal to identity (1) becomes

$$x_i(t+1) = A_i x_i(t) + B_i u_i(t) + E_i U_i u_{i-1}(t),$$

and equivalently for $\overline{E}_i = E_i U_i$

$$x_i(t+1) = A_i x_i(t) + B_i u_i(t) + \overline{E}_i u_i(t).$$

We state the main result of this section.

Proposition 1: Consider n systems where each system i is described by

$$x_i(t+1) = A_i x_i(t) + B_i u_i(t) + E_i u_{i-1}(t).$$

Let $u_i^\star$ be the concatenation of $u_i^\star(t_0), \ldots, u_i^\star(t_0+\tau-1)$, $i = 1:n$. Then $u^\star = [u_1^\star, \ldots, u_n^\star]$ is the minimiser of (5), and

$$u_i^\star = -(G_i^\top Q_i G_i + R_i + \overline{R}_{i+1})^{-1} \left(G_i^\top Q_i H_i u_{i-1} + G_i^\top Q_i L_i \xi_i + r_{i+1}\right)$$

$$\overline{R}_i = H_i^\top \Big(Q_i - Q_i G_i (G_i^\top Q_i G_i + R_i + \overline{R}_{i+1})^{-1} G_i^\top Q_i\Big) H_i,$$

$$r_i = H_i^\top Q_i L_i \xi_i - H_i^\top Q_i G_i (G_i^\top Q_i G_i + R_i + \overline{R}_{i+1})^{-1} (G_i^\top Q_i L_i \xi_i + r_{i+1}),$$

where $\overline{R}_{n+1} = 0$ and $r_{n+1} = 0$.

Proof: The proof is a consequence of applying standard dynamic programming principle. First define

$$g_i(u_i, \ldots, u_n) = \sum_{l=i}^{n} \sum_{t=t_0+1}^{t_0+\tau} f_l(u_l(t))$$

Checking the KKT conditions for system n yields the optimal input that minimises $g_n(u_i)$:

$$u_n^\star = -(G_n^\top Q_n G_n + R_n)^{-1} G_n^\top Q_n (H_n u_{n-1} + L_n \xi_n).$$

Evaluating $g_n(u_n)$ at $u_n^\star$ we obtain

$$g_n(u_n)_{|u_n^\star} = u_{n-1}^\top \overline{R}_n u_{n-1} + 2 u_{n-1}^\top r_n + \gamma_n$$

where

$$\overline{R}_n = H_n^\top \Big(Q_n - Q_n G_n (G_n^\top Q_n G_n + R_n)^{-1} G_n^\top Q_n\Big) H_n,$$

$$r_n = H_n^\top \Big(Q_n - Q_n G_n (G_n^\top Q_n G_n + R_n)^{-1} G_n^\top Q_n\Big) L_n \xi_n,$$

and γ_n is a constant. Applying the dynamic programming principle we obtain

$$u_i^\star = -(G_i^\top Q_i G_i + R_i + \overline{R}_{i+1})^{-1} G_i^\top Q_i H_i u_{i-1} - (G_i^\top Q_i G_i + R_i + \overline{R}_{i+1})^{-1} (G_i^\top Q_i L_i \xi_i + r_{i+1})$$

$$\overline{R}_i = H_i^\top \Big(Q_i - Q_i G_i (G_i^\top Q_i G_i + R_i + \overline{R}_{i+1})^{-1} G_i^\top Q_i\Big) H_i$$

$$r_i = H_i^\top Q_i L_i \xi_i - H_i^\top Q_i G_i (G_i^\top Q_i G_i + R_i + \overline{R}_{i+1})^{-1} (G_i^\top Q_i L_i \xi_i + r_{i+1}).$$

■

Proposition 1 and its proof shed some light on the nature of calculations and information required to obtain the optimal control input to minimise (5). The optimal control input $u_i^\star$ can be constructed through access to the local values G_i, H_i, L_i, Q_i, R_i, and receiving values $\overline{R}_{i+1}$ and r_{i+1} from system $i+1$ and u_{i-1} from system $i-1$.

2) Coupling Through States: In this section we consider the case where $v_i(t) = X_i x_{i-1}(t)$. Similar to the previous section, without loss of generality, we assume X_i is identity. We state the main result of this section.

Proposition 2: Consider n systems where each system i is described by

$$x_i(t+1) = A_i x_i(t) + B_i u_i(t) + E_i x_{i-1}(t).$$

Let $u_i^\star$ be the concatenation of $u_i^\star(t_0), \ldots, u_i^\star(t_0+\tau-1)$, $i = 1:n$. Then $u^\star = [u_1^\star, \ldots, u_n^\star]$ is the minimiser of (5), and

$$u_i^\star = -(G_i^\top \widetilde{Q}_i G_i + R_i)^{-1} \left(G_i^\top \widetilde{Q}_i H_i x_{i-1} + G_i^\top \widetilde{Q}_i L_i \xi_i + G_i^\top q_{i+1}\right)$$

$$\overline{Q}_i = H_i^\top \Big(\widetilde{Q}_i - \widetilde{Q}_i G_i (G_i^\top \widetilde{Q}_i G_i + R_i)^{-1} G_i^\top \widetilde{Q}_i\Big) H_i,$$

$$\widetilde{Q}_i = Q_i + \overline{Q}_{i+1},$$

$$q_i = H_i^\top \Big(\widetilde{Q}_i - \widetilde{Q}_i G_i (G_i^\top \widetilde{Q}_i G_i + R_i)^{-1} G_i^\top \widetilde{Q}_i\Big) L_i \xi_i + \Big(H_i^\top - H_i^\top \widetilde{Q}_i G_i (G_i^\top \widetilde{Q}_i G_i + R_i)^{-1} G_i^\top \widetilde{Q}_i\Big) q_{i+1},$$

where $\widetilde{Q}_n = Q_i$ and $q_n = 0$

Proof: Similar to the proof of Proposition 1 the proof is a consequence of applying standard dynamic programming principle. We have omitted the proof for the sake of brevity.

■

Again similar to the previous case, the optimal control input $u_i^\star$ can be constructed through access to the local values G_i, H_i, L_i, Q_i, R_i, and receiving values $\overline{Q}_{i+1}$ and q_{i+1} from system $i+1$ and x_{i-1} from system $i-1$.

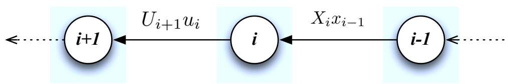

Fig. 2. An example of systems under heterogeneous coupling.

TABLE I
DIFFERENT COUPLINGS OF i.

	Case I	Case II	Case III	Case IV
v_{i+1}	$U_{i+1}u_i$	$X_{i+1}x_i$	$U_{i+1}u_i$	$X_{i+1}x_i$
v_i	$U_i u_{i-1}$	$X_{i+1}x_{i-1}$	$X_i x_{i-1}$	$U_i u_{i-1}$

3) Heterogeneous Coupling: We conclude this section by commenting on situations where the couplings in a system are not of the same type. An example for such scenarios is shown in Fig. 2 where $v_{i+1} = U_{i+1}u_i$ and $v_i = X_i x_{i-1}$. Different coupling arrangements for system i are described in Table I. Note that Cases I and II can be addressed by applying the results of the previous two subsection directly. In what follows we present the solution for calculating $u_i^\star$ for Cases III and IV.

Case III: : Let $g_{i+1}(u_{i+1}, \ldots, u_n)$ evaluated at $u_{i+1}^\star$ be

$$g_{i+1}(u_{i+1}, \ldots, u_n)_{|_{u_{i+1}^\star}} = u_i^\top \overline{R}_{i+1} u_i + 2u_i^\top r_{i+1} + \gamma_{i+1}.$$

Then it is easy to check

$$u_i^\star = -(G_i^\top Q_i G_i + R_i + \overline{R}_{i+1})^{-1} (G_i Q_i H_i x_{i-1} + G_i Q_i L_i \xi_i + r_{i+1}).$$

Moreover, $g_{i+1}(u_{i+1}, \ldots, u_n)$ evaluated at $u_i^\star$ is

$$g_{i+1}(u_{i+1}, \ldots, u_n)_{|_{u_{i+1}^\star}} = x_{i-1}^\top \overline{Q}_i x_{i-1} + 2x_{i-1}^\top q_i + \bar{\gamma}_i.$$

where

$$\overline{Q}_i = H_i^\top \Big(Q_i - Q_i G_i (G_i^\top Q_i G_i + R_i + \overline{R}_{i+1})^{-1} G_i^\top Q_i\Big) H_i$$

and

$$q_i = H_i^\top Q_i L_i \xi_i - H_i^\top Q_i G_i (G_i^\top Q_i G_i + R_i + \overline{R}_{i+1})^{-1} (G_i^\top Q_i L_i \xi_i + r_{i+1}).$$

Case IV: : Similar to the previous case, let $g_{i+1}(t_0, \tau)$ evaluated at $u_{i+1}^\star$ be

$$g_{i+1}(u_{i+1}, \ldots, u_n)_{|_{u_{i+1}^\star}} = x_i^\top \overline{Q}_{i+1} x_i + 2x_i^\top q_{i+1} + \gamma_{i+1}.$$

Then it is easy to check

$$u_i^\star = -(G_i^\top \widetilde{Q}_i G_i + R_i)^{-1} (G_i \widetilde{Q}_i H_i u_{i-1} + G_i \widetilde{Q}_i L_i \xi_i + G_i^\top q_{i+1}).$$

where $\widetilde{Q}_i = Q_i + \overline{Q}_{i+1}$. Moreover, $g_i(t_0, \tau)$ evaluated at $u_i^\star$ is

$$g_{i+1}(u_{i+1}, \ldots, u_n)_{|_{u_{i+1}^\star}} = u_{i-1}^\top \overline{R}_i u_{i-1} + 2u_{i-1}^\top r_i + \bar{\gamma}_i.$$

where

$$\overline{Q}_i = H_i^\top \Big(Q_i - \widetilde{Q}_i G_i (G_i^\top \widetilde{Q}_i G_i + R_i + \overline{R}_{i+1})^{-1} G_i^\top Q_i\Big) H_i$$

and

$$q_i = H_i^\top Q_i L_i \xi_i - H_i^\top Q_i G_i (G_i^\top Q_i G_i + R_i + \overline{R}_{i+1})^{-1} (G_i^\top Q_i L_i \xi_i + r_{i+1}).$$

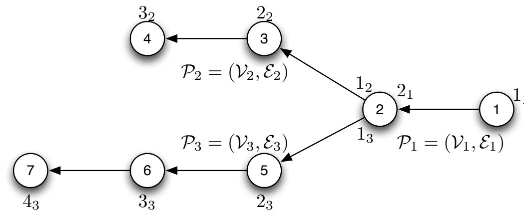

Fig. 3. A sample rooted tree and three paths with vertex sets $\mathcal{V}_1 = \{1_1, 2_1\}$, $\mathcal{V}_2 = \{1_2, 2_2, 2_3\}$, and $\mathcal{V}_3 = \{1_3, 2_3, 3_3, 4_3\}$.

B. Rooted Trees

The systems considered in Section III-A can be modelled as a directed path from 1 to n. We now consider the case where systems $1:n$ form a *rooted tree* with 1 as its root. A rooted tree is a directed graph such that the underlying undirected graph has no cycles and all the edges point away from the root, that is the vertex with in-degree equal to zero. The graph described above is also known as an *arborescence*. Each vertex with an out-degree more than 1 is called a *fork* and each vertex with an out-degree equal to zero is termed a *leaf*. We can partition the rooted graph into directed paths connecting a fork to a leaf, a fork to another fork, and the root to the first fork, all without going through an intermediate fork. For all the systems i forming a rooted tree graph $|\mathcal{N}_i^-| = 1$ and $|\mathcal{N}_i^+| \geq 1$. Fig. 3 depicts a possible interconnection topology and the corresponding paths. The problem of finding the optimal control signals to minimise (5) then becomes applying the results obtained in the previous section directly to the nodes that are not forks and indirectly to the forks. In the next two subsection we formalise the solution for two different types of coupling considered earlier.

1) Coupling Through Control Signals: Having in mind the same spatial take on applying the dynamic programming principle as in the previous section, we formally characterise the optimal control input of each system i for the case where the systems form a rooted tree in the following proposition.

Proposition 3: Consider n systems where each system i is described by

$$x_i(t+1) = A_i x_i(t) + B_i u_i(t) + E_i u_{i^-}(t).$$

where $i^- \in \mathcal{N}_i^-$. Let $u_i^\star$ be the concatenation of $u_i^\star(t_0), \ldots, u_i^\star(t_0+\tau-1)$, $i = 1:n$. Then $u^\star = [u_1^\star, \ldots, u_n^\star]$ is the minimiser of (5), and

$$u_i^\star = -(G_i^\top Q_i G_i + R_i + \widetilde{R}_i)^{-1} \Big(G_i^\top Q_i H_i u_{i^-} + G_i^\top Q_i L_i \xi_i + \tilde{r}_i\Big)$$

where $\widetilde{R}_i = \sum_{l \in \mathcal{N}_i^+} \overline{R}_l$, $\widetilde{r}_i = \sum_{l \in \mathcal{N}_i^+} r_l$. If i is not a leaf

$$\overline{R}_i = H_i^\top \Big(Q_i - Q_i G_i (G_i^\top Q_i G_i + R_i + \widetilde{R}_i)^{-1} G_i^\top Q_i\Big) H_i,$$

$$r_i = H_i^\top Q_i L_i \xi_i - H_i^\top Q_i G_i (G_i^\top Q_i G_i + R_i + \widetilde{R}_i)^{-1} (G_i^\top Q_i L_i \xi_i + \widetilde{r}_i)$$

and if i is a leaf $\widetilde{R}_i = 0$, and $\widetilde{r}_i = 0$.

2) Coupling Through States: For the case where the coupling is through states we have the following result.

Proposition 4: Consider n systems where each system i is described by

$$x_i(t+1) = A_i x_i(t) + B_i u_i(t) + E_i x_{i^-}(t).$$

where $i^- \in \mathcal{N}_i^-$. Let $u_i^\star$ be the concatenation of $u_i^\star(t_0), \ldots, u_i^\star(t_0+\tau-1)$, $i = 1:n$. Then $u^\star = [u_1^\star, \ldots, u_n^\star]$ is the minimiser of (5), and

$$u_i^\star = -(G_i^\top \widetilde{Q}_i G_i + R_i)^{-1} \big(G_i^\top \widetilde{Q}_i H_i x_{i^-} + G_i^\top \widetilde{Q}_i L_i \xi_i + G_i^\top \widetilde{q}_i\big)$$

where $\widetilde{Q}_i = Q_i + \sum_{l\in\mathcal{N}_i^+} \overline{Q}_l$, $\widetilde{q}_i = \sum_{l\in\mathcal{N}_i^+} q_l$. If i is not a leaf $\overline{Q}_i = H_i^\top \Big(\widetilde{Q}_i - \widetilde{Q}_i G_i (G_i^\top \widetilde{Q}_i G_i + R_i)^{-1} G_i^\top \widetilde{Q}_i\Big) H_i$,

$$\begin{aligned} q_i =& H_i^\top \Big(\widetilde{Q}_i - \widetilde{Q}_i G_i (G_i^\top \widetilde{Q}_i G_i + R_i)^{-1} G_i^\top \widetilde{Q}_i\Big) L_i \xi_i \\ &+ \Big(H_i^\top - H_i^\top \widetilde{Q}_i G_i (G_i^\top \widetilde{Q}_i G_i + R_i)^{-1} G_i^\top \widetilde{Q}_i\Big) q_{i+1}, \end{aligned}$$

and if i is a leaf $\widetilde{Q}_i = Q_i$, and $\widetilde{q}_i = 0$.

3) Heterogeneous Coupling: One can consider the case where not all the inter-system couplings are of the same nature. For example the case where a fork system i has two out-neighbours, $\mathcal{N}_j^+ = \{j, l\}$, and influences one of them through its input signal and other one through its state, i.e., $v_j(t) = U_j u_i(t)$ and $v_l(t) = X_l x_i(t)$. The procedure to compute the optimal input signal at each of the systems, while not identical, is similar to the ones introduced earlier and we omit them for brevity.

C. η-Neighbour Graph

First, we define the graph that describes the network we consider in this section. We call the directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ an *η-neighbour graph* if $\mathcal{N}_i^- = \{i-1, \ldots, \max(1, i-\eta)\}$ for a positive integer η. Assuming that the interconnection graph is an η-neighbour graph we address Problem 1 in this section. For the sake of brevity, we consider the case where the inter-system couplings are achieved via control inputs, for the interconnection graph described above it is easy to check that each system i is described by

$$x_i(t+1) = A_i x_i(t) + B_i u_i(t) + E_i \Big(\sum_{l=1}^{|\mathcal{N}_i^-|} U_{i,i-l} u_{i-l}(t)\Big).$$

To be able to address Problem 1 we form $\bar{n} = \lceil n/\eta \rceil$ augmented systems. If $n \mod \eta \neq 0$ then each system $j \geq 2$ is described by

$$\bar{x}_j(t+1) = \overline{A}_j \bar{x}_j(t) + \overline{B}_j \bar{u}_j(t) + \overline{E}_j \bar{u}_{j-1}(t) \qquad (6)$$

where $\bar{x}_j(t) = [x_i(t)^\top, \ldots, x_{i+\eta-1}(t)^\top]^\top$, $\bar{u}_j(t) = [u_i(t)^\top, \ldots, u_{i+\eta-1}(t)^\top]^\top$, $\overline{A}_j = \text{diag}(A_i, \ldots, A_{i+\eta-1})$,

$$\overline{B}_j = \begin{bmatrix} B_i & 0 & \ldots & 0 \\ E_{i+1}U_{i+1,i} & B_{i+1} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ E_{i+\eta-1}U_{i+\eta-1,i} & E_{i+\eta-1}U_{i+\eta-1,i+1} & \ldots & B_{i+\eta-1} \end{bmatrix},$$

and

$$\overline{E}_j = \begin{bmatrix} E_i U_{i,i-\eta} & E_i U_{i,i-\eta+1} & \ldots & E_i U_{i,i-1} \\ 0 & E_{i+1}U_{i+1,i-\eta+1} & \ldots & E_{i+1}U_{i+1,i-1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & E_{i+\eta-1}U_{i+\eta-1,i-1} \end{bmatrix}.$$

For the augmented system 1 we have $\bar{x}_1(t) = [x_1(t)^\top, \ldots, x_\rho(t)^\top]^\top$, $\bar{u}_1(t) = [u_1(t)^\top, \ldots, u_\rho(t)^\top]^\top$, $\overline{A}_1 = \text{diag}(A_1, \ldots, A_\rho)$,

$$\overline{B}_j = \begin{bmatrix} B_1 & 0 & \ldots & 0 \\ E_2U_{2,1} & B_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ E_\rho U_{\rho,1} & E_\rho U_{\rho,2} & \ldots & B_\rho \end{bmatrix},$$

and $\overline{E}_1 = 0$. For the case where $n \mod \eta = 0$, (6) describes all $j = 1:\bar{n}$ augmented systems. Assuming that all agents i in an augmented system j as described above have a local model of the augmented system j, i.e. exchange information with all the other systems in j, and receive information from the augmented system $j+1$ one can apply the results from Section III-A to obtain $u_i^\star$ for $i = 1:n$ that minimise (5). Following similar steps one can compute the optimal control input for the case where the systems are coupled via state variables.

D. Rooted Cacti

We start this section by defining the class of networks we consider here. An undirected graph is a *cactus* if any two cycles have at most one common vertex. We extended this definition to directed graphs in what follows. We call an acyclic directed graph a *rooted cactus* if there is only one vertex with zero in-neighbours and if the underlying undirected graph is a cactus. An example of a rooted cactus is depicted in Fig. 4(a). We form $\bar{n}$ augmented systems as described in what follows. All systems i that belong to the same cycle in the underlying undirected graph are considered to form one augmented system. For the case where a system belong to two undirected graphs, it is considered to be a part of the augmented system where its in-neighbours are a member of. Moreover, if a system is not a member of a cycle in the undirected graph then it does not form a larger augment system with any other system. Additionally, it is assumed that the systems form an augmented system have full access to the other systems forming the same augmented system. An example for which systems form augmented systems is given in Fig. 4(b). The interconnection graph after forming the augmented systems is depicted in Fig. 4(c). It is easy to show that after forming the augmented systems described above the network of such augmented systems is isomorphic to a rooted tree. Thus, one can apply the results presented in Section III-B to solve Problem 1.

IV. Numerical Examples

Initially we consider 20 coupled systems where

$$x_i(t+1) = Ax_i(t) + Bu_i(t) + Eu_{i-1}(t), \quad i = 1:n$$

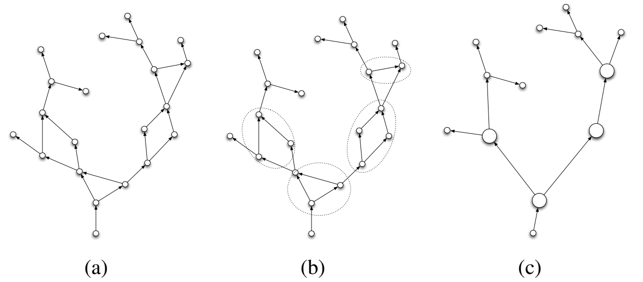

Fig. 4. An example for a rooted cactus network, and augmented systems.

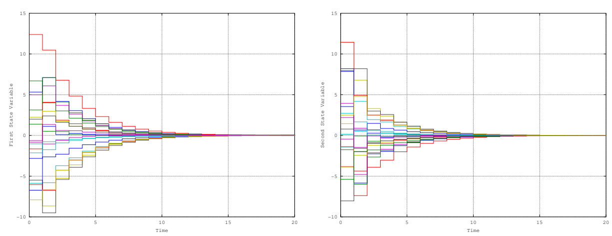

Fig. 5. The trajectories of 20 systems coupled through control inputs.

where $A = \begin{bmatrix} 1 & 0.5 \\ 1 & 0 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $E = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}$, and $u_0(t) = 0$. Additionally, $Q_i(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $R_i(t) = 1$, and $\tau = 20$, $i = 1 : n$, $t = t_0 : t_0 + \tau$. The state trajectories of the systems after applying the inputs computed as described in Proposition 1, coincide with the trajectories obtained form solving the problem in a direct way without exploiting structure, to minimise (5) as depicted in Fig. 5. In the next scenario to demonstrate the efficiency of the method outlined here we compare the average time required to calculate the optimal inputs, for paths of 10 different lengths – from 10 to 100. The average time required for all the systems to calculate their optimal inputs verses the length (size) of the path is depicted in Fig. 6. It can be seen that the method proposed here computes the optimal control inputs more than 10 times faster for the smaller number of systems and 100 times faster for longer paths than a direct computation where a global static optimisation problem over the finite-horizon is solved. To conform with the guidelines of reproducible research the `Octave/Matlab` file to generate the results presented above can be found at `http://eemensch.tumblr.com/d_lqr_dp`.

V. Concluding Remarks

In this paper, we proposed a computationally efficient method to calculate optimal control inputs in a networks of linear systems with underlying acyclic graphs and made comparisons with other types of solutions where a global problem needs to be solved at each system. We note that the results obtained here are immediately applicable to the scenarios where the signals are corrupted by i.i.d. noise variables. Moreover, the spatial decomposition idea introduced is applicable to the case where the systems have nonlinear dynamics, or in the presence of constraints albeit not so explicitly. This in turn facilitates applying the computationally

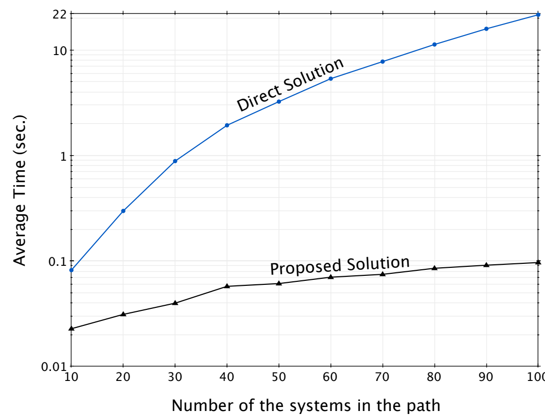

Fig. 6. The average time required to calculate the control inputs for the paths of different lengths using the proposed distributed scheme and the case where the global problem is solved.

efficient methods proposed here to address the problem of distributed model predictive control in the networks with the same classes of underlying graphs. For a recent review of the computational aspects of distributed MPC one can refer to [14].

References

[1] P. Järventausta, S. Repo, A. Rautiainen, and J. Partanen, "Smart grid power system control in distributed generation environment," *Annual Reviews in Control*, vol. 34, no. 2, pp. 277–286, 2010.

[2] M. Cantoni, E. Weyer, Y. Li, S. Ooi, I. Mareels, and M. Ryan, "Control of large-scale irrigation networks," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 75–91, 2007.

[3] A. Cárdenas, S. Amin, and S. Sastry, "Research challenges for the security of control systems," in *Proceedings of the 3rd conference on Hot topics in security*. USENIX Association, 2008, pp. 1–6.

[4] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *Control Systems Technology, IEEE Transactions on*, vol. 18, no. 2, pp. 267–278, 2010.

[5] E. Camponogara, D. Jia, B. Krogh, and S. Talukdar, "Distributed model predictive control," *Control Systems, IEEE*, vol. 22, no. 1, pp. 44–52, 2002.

[6] R. Scattolini, "Architectures for distributed and hierarchical model predictive control–a review," *Journal of Process Control*, vol. 19, no. 5, pp. 723–731, 2009.

[7] J. Rice and M. Verhaegen, "Distributed control: A sequentially semi-separable approach for spatially heterogeneous linear systems," *Automatic Control, IEEE Transactions on*, vol. 54, no. 6, pp. 1270–1283, 2009.

[8] J. K. Rice and M. Verhaegen, "Distributed computations and control in multi-agent systems," in *Autonomous Robots and Agents, 2009. ICARA 2009. 4th International Conference on*. IEEE, 2009, pp. 44–49.

[9] A. Rantzer, "Dynamic dual decomposition for distributed control," in *American Control Conference, 2009. ACC'09*. IEEE, 2009, pp. 884–888.

[10] L. Soltanian and M. Cantoni, "A 2-d roesser model for automated irrigation channels and tools for practical stability and performance analysis," in *Australian Control Conference*, 2012.

[11] Å. Holmgren, "Using graph models to analyze the vulnerability of electric power networks," *Risk analysis*, vol. 26, no. 4, pp. 955–969, 2006.

[12] B. Paten, M. Diekhans, D. Earl, J. John, J. Ma, B. Suh, and D. Haussler, "Cactus graphs for genome comparisons," *Journal of Computational Biology*, vol. 18, no. 3, pp. 469–481, 2011.

[13] D. Dimarogonas and K. Johansson, "Stability analysis for multi-agent systems using the incidence matrix: quantized communication and formation control," *Automatica*, vol. 46, no. 4, pp. 695–700, 2010.

[14] C. Conte, T. H. Summers, M. N. Zeilinger, M. Morari, and C. N. Jones, "Computational aspects of distributed optimization in model predictive control," in *IEEE Conference on Decision and Control*, 2012.