Optimal Distributed Controllers Realizable Over Arbitrary Networks

Andalam Satya Mohan Vamsi and Nicola Elia, *Senior Member, IEEE*

***Abstract*—In this paper, we provide a general networked controller design methodology for networked plants and apply it to solve the optimal $\mathcal{H}_2$ networked control problem. Both the plant and the controller are interconnected systems, interacting over the same arbitrary directed network with noiseless and delay-free communication links. We introduce the notions of network implementability and network realizability and analyze the structure of network implementable and realizable systems. Based on the structural property of network implementable systems, under certain network-related constraints, we characterize the set of all stabilizing controllers that are implementable over the given network using the state-space version of Youla parametrization. Moreover, we provide a constructive procedure to implement the controllers as sub-systems interacting over the given network without affecting the stability of the feedback networked system. The distributed $\mathcal{H}_2$ control problem is then cast as a convex optimization problem and its solution is shown to provide the optimal distributed controller over the given network in terms of its network interacting components. The results of this paper allow one to apply many classical results and approaches of multivariable robust control theory to networked systems.**

***Index Terms*—Distributed control, network realizability theory, networked control systems, optimal $\mathcal{H}_2$ control.**

I. Introduction

A networked, or distributed, or interconnected system is a group of plants or sub-systems interacting over a communication network. With the increasing number of applications in the field of networked systems, there has been a great surge in research towards networked controller design for such systems.

One of the main objectives of this research is to find optimal networked controllers for networked plants and to provide the constituent sub-systems of the designed controller. The controllers need to be implementable over the existing communication network. This setting is reasonable in practice when the network infrastructure of the plant is already in place, which is often the case, or when networked plants and networked controllers need to be built simultaneously on a given network.

In this paper, as an example of the general methodology we propose, we solve the optimal networked $\mathcal{H}_2$ control problem for a large class of linear time-invariant discrete-time networked systems.

Manuscript received July 5, 2012; revised July 15, 2014 and March 31, 2015; accepted April 14, 2015. Date of publication May 15, 2015; date of current version December 24, 2015. This work was supported in part by the National Science Foundation (NSF) under Grants ECCS0901846 and CNS1239319 and by the Air Force Office of Scientific Research (AFOSR) under Grant FA95501510119. Recommended by Associate Editor H. Ishii.

A preliminary and partial version of this paper has appeared in [1]. Some of the results of this paper have been summarized in a tutorial form and without proofs as part of a book chapter [2].

The authors are with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011 USA (e-mail: vamsi68@iastate.edu; nelia@iastate.edu).

Digital Object Identifier 10.1109/TAC.2015.2434051

Our approach applies to general networked systems composed by heterogenous sub-systems interacting over arbitrary network topologies. Each subsystem only uses local information but does not instantaneously relay its local inputs to its neighbors. This requirement captures the practical aspect that the communication over the network is not instantaneous and focuses the development on the network topology, eliminating confusion between network topology and information patterns.

The recent literature on distributed controller synthesis has analyzed various classes of easily searchable structured systems in the transfer function and in the state-space domain. Spatially invariant systems were studied in [2]–[7]. [8], [9] considered systems with triangular and band structures, [10] focused on symmetrically interconnected systems, [11] considered poset-causal systems. System structures satisfying quadratic invariance property were studied in [12], [13] and [14], which has lifted the restriction requiring a stable stabilizing central controller and provided a complete parametrization of all the stabilizing controllers with such structures. Systems over general graphs were considered by [15], identical sub-systems connected over a graph with diagonalizable "pattern matrix" were considered by [16] and heterogeneous sub-systems connected over arbitrary undirected graphs were considered by [17].

Networked systems and structured systems have become so intertwined that they are often identified with each other. However, and this is the starting point of the paper, it is not always clear if and how a structured representation, consistent with a given network, represents a networked system composed of subsystems interacting over a network.

In the state-space approaches, the conditions for realizability over the given network can be easily specified [17]. Thus, if a controller is designed with state-space matrices following the desired sparsity constraints, it can be easily realized as subsystems interacting over the given network. However, optimization methods based on searching state-space structures tend to be sub-optimal [16]–[18], due to the network constraints and the variety of multi-objective specifications, which make the optimal controller order unknown or difficult to know *a priori*. Recent important exceptions are optimal state-space characterizations for special cases specific to $\mathcal{H}_2$ cost: [19] provides the optimal state space controller for two-player triangular structures; while, [20] uses dynamic programming to provide the optimal networked controller over strongly connected networks, with network settings similar to ours but in the case of state-feedback. Our approach, instead, applies to output feedback, networks not necessarily strongly connected and other performance measures as well, e.g., ℓ_1.

Input-output approaches are better suited when looking for optimal networked controllers as they do not restrict the controller order and thus more naturally handle multi-objective problems and network constraints. However, in general, it is not clear how to realize a controller designed as a structured transfer function matrix, in terms of local controllers exchanging information over the given network.

Our methodology merges the state-space approach with an input-output approach, combining their respective benefits. Our development is centered on finding representations of networked systems that are easily searchable. These representations need to be rich enough to guarantee an easy solution to physical implementation of the structured system as a set of subsystems interacting over the network.

In order to do so, we develop specific aspects of system theory for networked systems pertaining to implementation and realization theory, otherwise well understood for unstructured LTI systems [21].

We first introduce the notion of network implementability for systems with structured state-spaces. The implementation of a networked system needs to be free from unobservable and uncontrollable unstable modes. Based on implementability, we propose the notion of network realizability for systems with transfer functions having certain delay and sparsity structures. The realization problem for LTI systems is about finding a (minimal) state-space realization consistent with a given transfer function or impulse response [21]. Finding a networked realization is more difficult and apparently the problem has not been studied. In fact, there are transfer function matrices, consistent with a given graph, that are indeed not network realizable according to our definition, as shown in [22], motivated by [1]. In particular, the question of network realizability of optimal structured controllers has not been explicitly addressed in the literature until [1], [23]. It has been implicitly guaranteed in some special structured problems [9], [16]. However, no explicit or general network realization procedures are known.

In this paper, we show that stable input-output representations consistent with the interconnection graph are always network realizable and provide a realization procedure. We use this result and our networked system representation to obtain a parametrization of all the network stabilizing controllers implementable over the given network, under certain conditions, and provide a procedure to implement them. We finally apply the approach to solve the networked $\mathcal{H}_2$ problem and present an example, where explicit networked sub-controller are derived.

In summary, based on [1], [18], [23], this paper is among the first to formulate the problem of networked system realization. It also proposes a general networked controller design methodology applicable to a very large class of networked systems, and provides a networked realization of the optimal controller.

II. Notation

A. Graph Model

In this paper, we deal with networked systems that are best described using a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{1, \ldots, n\}$ represents the nodes/vertices of the graph or the subsystems in the network, and $\mathcal{E} \subset \mathcal{V}^2$ represents the edge-set or the set of communication links between different sub-systems. We say, edge $(j, i) \in \mathcal{E}$ if there exists a directed link from node j to node i.

For ease of notation, we consider $(i, i) \in \mathcal{E}\, \forall i \in \mathcal{V}$, i.e., we allow self-loops at all the nodes of the graph. The first vertex j in the edge (j, i) is called its *tail* and the second vertex i is its *head*.

A *walk* in $\mathcal{G}$ is an alternating sequence $v_1 e_1 v_2 e_2 \ldots e_{k-1} v_k$ of vertices $v_i \in \mathcal{V}$ and edges $e_j \in \mathcal{E}$ such that the tail of e_i is v_i and the head of e_i is v_{i+1} for every $i = 1, 2, \ldots, k-1$. To simplify the notation, since we assume unique directed edges between nodes, we write the walk only as a sequence of the vertices as $v_1 v_2 \ldots v_k$. A *path* is a walk where all the vertices are distinct. *Length* of a walk is defined as the number of edges in the walk. A *shortest path* from node j to node i $(j \neq i)$ is defined as a path from j to i with shortest length.

Given a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, define the adjacency matrix $\mathcal{A}(\mathcal{G})$ to be a binary matrix such that

$$[\mathcal{A}(\mathcal{G})]_{ij} = \begin{cases} 1 & \text{if } (j, i) \in \mathcal{E} \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Similarly, define an m-step adjacency matrix $\mathcal{A}_m(\mathcal{G})$ to be a binary matrix such that

$$[\mathcal{A}_m(\mathcal{G})]_{ij} = \begin{cases} 1 & \text{if there exists at least one walk} \\ & \text{from node } j \text{ to node } i \text{ of length} \leq m \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Let the shortest path length from node j to node i be denoted by $l(j, i)$. From (2), it is easy to see that the shortest path length from node j to node i $(j \neq i)$ is given by

$$l(j, i) = \inf \left\{ m : [\mathcal{A}_m(\mathcal{G})]_{ij} \neq 0 \right\}. \quad (3)$$

For notational convenience, we define $l(j, i) = \infty$ if there is no path from j to i, and we abuse the terminology by defining $l(j, j) = 0$.

We will often use the adjacency matrix of $\mathcal{G}$-without self-loops. We denote it by $\tilde{\mathcal{A}}(\mathcal{G}) \triangleq \mathcal{A}(\mathcal{G}) - I$. Define directed neighborhoods around each node i, the in-neighbors $\mathcal{N}_i^- = \{j | (j, i) \in \mathcal{E}\}$ and the out-neighbors $\mathcal{N}_i^+ = \{j | (i, j) \in \mathcal{E}\}$, which are the sets of nodes that have edges to and from node i. With a slight abuse of terminology, we shall refer to the network by $\mathcal{G}$ as well as the underlying graph representing the network.

B. General

We refer to a column-vector as *vector*. To make representations compact, we use the notation $\mathbf{vert}[x_i]_{i \in \mathcal{I}}$ and $\mathbf{hor}[x_i]_{i \in \mathcal{I}}$ for vertical and horizontal concatenation of vectors or matrices $\{x_i\}_{i \in \mathcal{I}}$, of appropriate dimension, where $\mathcal{I}$ is an index set. Let $[x_{ij}]_{i, j \in \mathcal{I}}$ represent a matrix formed by arranging the sub-matrices $\{x_{ij}\}_{i,j}$ as $\mathbf{vert}[\mathbf{hor}[x_{ij}]_{j \in \mathcal{I}}]_{i \in \mathcal{I}}$. Also, let $\mathbf{diag}[x_i]_{i \in \mathcal{I}}$ denote the matrix formed by arranging the vectors or matrices $\{x_i\}_{i \in \mathcal{I}}$ in a block diagonal fashion and the remaining entries being zeros. Sometimes, if the index set $\mathcal{I}$ equals $\{1, \ldots, n\}$, then we will not explicitly mention the index set. Given a matrix $A = [a_1, \ldots, a_n] \in \mathbb{C}^{m \times n}$, where $\{a_i\}_i$ denote the columns of A, we associate a vector $\mathbf{vec}(A) = \mathbf{vert}[a_i]_i \in \mathbb{C}^{mn}$ which is a vector formed by vertically concatenating the columns of matrix A. Define $\mathbf{vec}^{-1}(\cdot)$ as the inverse operation of the $\mathbf{vec}(\cdot)$ such that $\mathbf{vec}^{-1}(\mathbf{vec}(A)) = A$. When required, we shall use I for an identity matrix and 0 for a zero matrix of appropriate size.

In this paper, we will come across block matrices that are made up of smaller sub-matrices. These matrices are best described in terms of their sparsity structures.

Definition 1: We say a block matrix $A = [A_{ij}]_{i,j\in\{1,\ldots,n\}}$ is *structured according to* an $n \times n$ binary matrix J if the sub-matrix A_{ij} is a zero matrix whenever $J_{ij} = 0$. The dimensions of the sub-matrices $\{A_{ij}\}_{i,j}$ are described using two integer-valued vectors as follows. Let $\mathcal{P}_a = (a_1, \ldots, a_n)$ and $\mathcal{P}_b = (b_1, \ldots, b_n)$ be two n-tuples with a_i and b_i being integers for all $i \in \{1, \ldots, n\}$. Then, matrix A is said to be *partitioned according to* $(\mathcal{P}_a, \mathcal{P}_b)$ if the sub-matrix A_{ij} has dimensions $a_i \times b_j \forall i, j$.

This definition of partitioning is easily extended to the case of vectors too. A vector x is said to be *partitioned according to* $\mathcal{P}_a$ if it can be written as $\mathbf{vert}[x_i]_{i\in\{1,\ldots,n\}}$ where x_i is a real vector of size a_i for all $i \in \{1, \ldots, n\}$.

For example, according to the above definitions, the following matrix:

$$A = \left[\begin{array}{c|cc|ccc} 1 & 0 & 1 & 0 & 0 & 0 \\ \hline 3 & 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 & 0 \\ \hline 0 & 3 & 0 & 2 & 1 & 2 \end{array}\right]$$

is structured according to a binary matrix $J = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$ and partitioned according to $(\mathcal{P}_a, \mathcal{P}_b)$ where $\mathcal{P}_a = (1, 2, 1)$ and $\mathcal{P}_b = (1, 2, 3)$. Note that blocks in A corresponding to 0's in J must be identically zero, while blocks in A corresponding to 1's in J can have zeros.

The following two lemmas are used in the later part of this paper to describe properties of state-space and input-output representations of interconnected systems.

Lemma 1: Let J be an $n \times n$ binary matrix and $\mathcal{P}_a$, $\mathcal{P}_b$, $\mathcal{P}_c$, $\mathcal{P}_d$ be n-tuples. If matrices E, F, G are partitioned according to $(\mathcal{P}_a, \mathcal{P}_b)$, $(\mathcal{P}_b, \mathcal{P}_c)$ and $(\mathcal{P}_c, \mathcal{P}_d)$, respectively, and E, G are block diagonal while F is structured according to J; then EFG is structured according to J and partitioned according to $(\mathcal{P}_a, \mathcal{P}_d)$.

Proof: From the hypothesis, we see that $E = [E_{ij}]_{i,j}$, $F = [F_{ij}]_{i,j}$ and $G = [G_{ij}]_{i,j}$ where E_{ij} and G_{ij} are zero matrices when $i \neq j$ while $F_{ij} = 0$ when $J_{ij} = 0$. From the properties of block matrices and matrix multiplication, it is easy to see that EFG is a block matrix which is partitioned according to $(\mathcal{P}_a, \mathcal{P}_d)$. Thus, we can write $EFG = [H_{ij}]_{i,j}$ in terms of some sub-matrices H_{ij} which have dimensions $\mathcal{P}_i^a \times \mathcal{P}_j^d$. Thus

$$H_{ij} = \sum_{k=1}^{n}\sum_{m=1}^{n} E_{ik}F_{km}G_{mj} = \sum_{m=1}^{n} E_{ii}F_{im}G_{mj} = E_{ii}F_{ij}G_{jj} \tag{4}$$

since $E_{ik} = 0 \, \forall i \neq k$ and $G_{mj} = 0 \, \forall m \neq j$. From (4), we see that $H_{ij} = 0$ whenever $J_{ij} = 0$ since $F_{ij} = 0$ whenever $J_{ij} = 0$. Thus, EFG is structured according to J and partitioned according to $(\mathcal{P}_a, \mathcal{P}_d)$. ∎

Lemma 2: Given a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with an adjacency matrix $\mathcal{A}(\mathcal{G})$, let $\{A_k\}_k$ be a sequence of block matrices that are structured according to $\mathcal{A}(\mathcal{G})$. Then $B_m = \prod_{k=1}^{m} A_k$ is structured according to $\mathcal{A}_m(\mathcal{G})$ for all m.

Proof: From the definition of $\mathcal{A}_m(\mathcal{G})$ in (2), we can see that $\mathcal{A}_1(\mathcal{G}) = \mathcal{A}(\mathcal{G})$. Thus from hypothesis, we know that $B_1 = A_1$ is structured according to $\mathcal{A}(\mathcal{G})$.

Now, assume that $B_m = \prod_{k=1}^{m} A_k$ is structured according to $\mathcal{A}_m(\mathcal{G})$ for some $m = p$. Using the matrix multiplication results for block matrices, we have $[B_{m+1}]_{ij} = \sum_{k=1}^{n}[A_{m+1}]_{ik}[B_m]_{kj}$. If there is no walk from node j to node i of length $m+1$, then, for any $k \in \mathcal{V}$, either there is no walk from node j to node k, in m steps or there is no directed edge from node k to node i. Thus, either $[B_m]_{kj}$ or $[A_{m+1}]_{ik}$ are zero-matrices for all k when $[\mathcal{A}_{m+1}(\mathcal{G})]_{ij} = 0$. Thus, $[B_{m+1}]_{ij}$ is a zero matrix when $[\mathcal{A}_{m+1}(\mathcal{G})]_{ij} = 0$, which implies that B_{m+1} is structured according to $\mathcal{A}_{m+1}(\mathcal{G})$. Thus, the given statement is true by mathematical induction. ∎

C. Structured Systems

In this paper, we consider (causal) Finite Dimensional Linear Time Invariant Discrete-Time (FDLTI-DT) systems. An FDLTI-DT system P with p inputs and q outputs can represented by a state-space model, $P = (A, B_u, C_y, D_{yu})$ where A, B_u, C_y, D_{yu} are matrices of compatible dimensions. P can also be represented as an input-output operator by its $q \times p$ impulse response sequence, $P = \{h(k)\}_0^\infty$ where $h(k) \in \mathbb{R}^{q\times p}$, or more conveniently by its transfer function matrix $P(z) = \sum_{k=0}^{\infty} h(k)z^{-k}$, the Z-transform of the impulse response, well-defined in the appropriate Region of Convergence. We use a function $\mathbf{tf}(P)$ to represent the unique transfer function matrix corresponding to any system P represented by its state-space equations or differential equations

$$P(z) = D_{yu} + \sum_{k=0}^{\infty} C_y A^k B_u z^{-k-1}. \tag{5}$$

Let $\mathcal{R}_p$ denote the set of real-rational proper transfer function matrices, $\mathcal{R}_{sp}$ denote the set of real-rational strictly-proper transfer function matrices and $\mathcal{RH}_\infty$ denote the set of real-rational proper stable transfer function matrices.

The inverse problem of finding a (minimal) set of difference equations or a state-space realization for a given transfer function matrix $P(z)$, is called realization problem [21]. While this problem is solved for general FDLTI systems, it is less understood when the realization must consist of dynamic difference equations distributed over several nodes interconnected over a network, as we will see later.

We next introduce FDLTI-DT systems that have structures in their state-space or input-output representations consistent with the sparsity structure induced by a graph defined earlier. These structures are amenable to be efficiently searched and, as we will see in the next section, capture the representations of systems over networks. However, a key question for us would be to find a networked system corresponding to a given structured system.

Definition 2: Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and the partitions $\mathcal{P}_x$, $\mathcal{P}_u$ and $\mathcal{P}_y$, let $\mathfrak{S}(\mathcal{G}, \mathcal{P}_x, \mathcal{P}_u, \mathcal{P}_y)$ denote the set of state-space realizations (A, B_u, C_y, D_{yu}) where A, C_y are structured according to the adjacency matrix $\mathcal{A}(\mathcal{G})$ given by (1), while B_u, D_{yu} are block-diagonal and the state-space matrices are partitioned as follows $A - (\mathcal{P}_x, \mathcal{P}_x)$, $B_u - (\mathcal{P}_x, \mathcal{P}_u)$, $C_y - (\mathcal{P}_y, \mathcal{P}_x)$, $D_{yu} - (\mathcal{P}_y, \mathcal{P}_u)$.

Definition 3: Given a graph $\mathcal{G}$ and the input and output partitions, $\mathcal{P}_u$ and $\mathcal{P}_y$, let $\mathfrak{T}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$ denote the set of transfer function matrices $P(z) = [P_{ij}(z)]_{i,j}$ that are partitioned according to $(\mathcal{P}_y, \mathcal{P}_u)$ where $P_{ij}(z)$ is of the form

$$P_{ij}(z) = \begin{cases} z^{-l(j,i)} H_{ij}(z) & \text{if } l(j,i) < \infty \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $l(j,i)$ is given by (3) extended to 0 and ∞ and $H_{ij}(z) \in \mathcal{R}_p$ for all i, j.

It is not surprising that Definition 2 and Definition 3 are consistent, in other words:

Lemma 3: If $P(z)$ denotes the input-output mapping from input vector $u(k)$ to output vector $y(k)$ corresponding to a system P with a state-space representation $(A, B_u, C_y, D_{yu}) \in \mathfrak{S}(\mathcal{G}, \mathcal{P}_x, \mathcal{P}_u, \mathcal{P}_y)$, then $P(z) \in \mathfrak{T}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$.

Proof: $P(z)$ is given by (5). From the partitions of state-space matrices, we see that $P(z)$ is partitioned according to $(\mathcal{P}_y, \mathcal{P}_u)$. Let $P(z) = [P_{ij}(z)]_{i,j}$, where $P_{ij}(z)$ is the transfer function matrix from input vector $u_j(k)$ to output vector $y_i(k)$. Note that $u(k) = \mathbf{vert}[u_i(k)]_i$ and $y(k) = \mathbf{vert}[y_i(k)]_i$ are partitioned according to $\mathcal{P}_u$ and $\mathcal{P}_y$, respectively. From Lemma 2 and the fact the B is block diagonal, it follows that $[CA^kB]_{ij} z^{-(k+1)}$ is zero if $0 \leq k < l(j,i) - 1$, for $l(j,i) \geq 1$. Thus, the transfer function matrix $P_{ij}(z)$ is of the form (6). ■

Remark 1: In the lack of further analysis, presented later, the reader should think of these structured representations as centralized systems, i.e., not necessarily representing a set of physically separated dynamical subsystems interconnected over the network. For example, consider the following system which is a discrete time version of the example proposed in [22]:

$$P(z) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{z-2} & \frac{1}{z-2} & 0 & 0 \\ \frac{1}{z-2} & \frac{1}{z-2} & 0 & 0 \end{bmatrix} \tag{7}$$

$P(z)$ has both sparsity and delay constraints consistent with the graph $\mathcal{G} = \{1 \to 3, 1 \to 4, 2 \to 3, 2 \to 4\}$, thus, $P(z) \in \mathfrak{T}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$ with $\mathcal{P}_y = \mathcal{P}_u = [1,1,1,1]$. A minimal realization of the above transfer function matrix is first order and given by

$$A = [2],\ B = [1,\ 1,\ 0,\ 0],\ C = [0,\ 0,\ 1,\ 1]',\ D = 0.$$

However this processing is not consistent with $\mathcal{G}$. The state cannot be at node 1 since there is no path from node 2 to obtain u_2. Similarly, the state cannot be at node 3, since there is no path from it to node 4 to send y_2. Same argument applies to node 2 and 4.

Another example is given by the following transfer function matrix:

$$P(z) = \begin{bmatrix} \frac{z-4}{z^2-7z+13} & -\frac{z-3}{z^2-7z+13} & \frac{1}{z^2-7z+13} \\ \frac{1}{z^2-7z+13} & \frac{z-4}{z^2-7z+13} & -\frac{z-3}{z^2-7z+13} \\ -\frac{z-3}{z^2-7z+13} & \frac{1}{z^2-7z+13} & \frac{z-4}{z^2-7z+13} \end{bmatrix} \tag{8}$$

which is in $\mathfrak{T}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$ with $\mathcal{P}_y = \mathcal{P}_u = [1,1,1]$, and $\mathcal{G} = \{3 \to 2 \to 1 \to 3\}$. It is easy to verify using Matlab, that the minimal state-space realization for the given transfer function is second order, with two complex conjugate eigenvalues at $-3.5 \pm i(\sqrt{3}/2)$. A minimal state-space realization is the following one with $D = 0$:

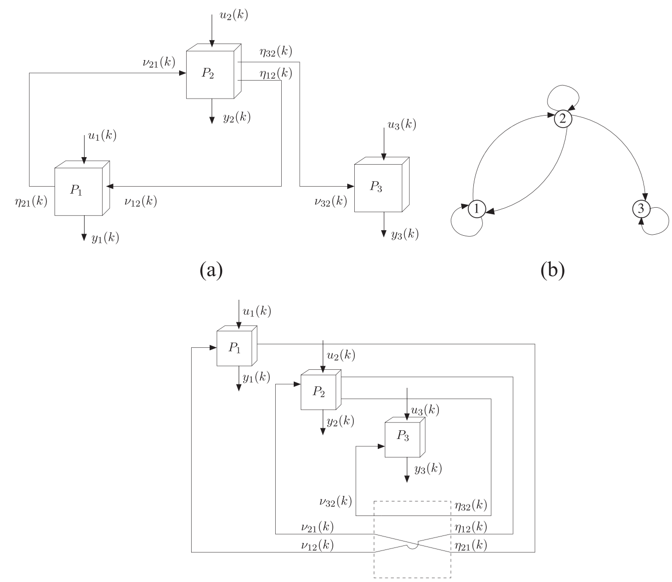

Fig. 1. A simple example of an interconnected system made of 3 different sub-systems and the underlying directed pseudograph representing the communication network. (a) An interconnected system. (b) Underlying pseudograph. (c) Interconnected system as an LFT of sub-systems and network.

$$A = \begin{bmatrix} 3.5 & \frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{2} & 3.5 \end{bmatrix}, \quad B = \begin{bmatrix} -\frac{1}{\sqrt{3}} & -\frac{\sqrt{3}-1}{2\sqrt{3}} & \frac{\sqrt{3}+1}{2\sqrt{3}} \\ -\frac{1}{\sqrt{3}} & \frac{\sqrt{3}+1}{2\sqrt{3}} & -\frac{\sqrt{3}-1}{2\sqrt{3}} \end{bmatrix}$$
$$C = \begin{bmatrix} -\frac{\sqrt{3}-1}{2} & -\frac{\sqrt{3}+1}{2} \\ -1 & 1 \\ \frac{\sqrt{3}+1}{2} & \frac{\sqrt{3}-1}{2} \end{bmatrix}.$$

This realization is consistent with a fusion center, which is in charge of the state evolution, it gathers the inputs, and determines the outputs.

In the next section, we introduce a large class of networked systems whose state-space and input-output representations belong to $\mathfrak{S}(\mathcal{G}, \mathcal{P}_x, \mathcal{P}_u, \mathcal{P}_y)$ and $\mathfrak{T}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$, respectively.

III. Networked Systems

In this section, we describe systems built over communication networks that we consider in this paper, and discuss some of the properties of their state-space and input-output descriptions.

A. Definition

A group of plants or sub-systems interacting over a communication network is termed as a *networked* or *interconnected system* (see Fig. 1). A networked system is characterized by 1) topology of the network; 2) local dynamics of the sub-systems; 3) properties of the interaction.

The topology of the network is described by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with n nodes. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with n nodes, we associate a sub-system $\{P_i\}_{i \in \{1,\ldots,n\}}$ to each node. The subsystems interact over communication links corresponding to the edges of the graph.

Each sub-system P_i is a FDLTI-DT system with local inputs $u_i(k)$, local outputs $y_i(k)$, network inputs $\nu_i(k)$ and network outputs $\eta_i(k)$. For the time being, each P_i has the following state-space description assumed to be minimal (to be further specified):

$$P_i : \begin{bmatrix} x_i(k+1) \\ y_i(k) \\ \eta_i(k) \end{bmatrix} = \begin{bmatrix} A_i & B_i^u & B_i^\nu \\ C_i^y & D_i^{yu} & D_i^{y\nu} \\ C_i^\eta & D_i^{\eta u} & D_i^{\eta\nu} \end{bmatrix} \begin{bmatrix} x_i(k) \\ u_i(k) \\ \nu_i(k) \end{bmatrix} \tag{9}$$

where $x_i(k)$ is the local state, and the following input-output description:

$$P_i(z) : \begin{bmatrix} Y_i(z) \\ \eta_i(z) \end{bmatrix} = \begin{bmatrix} P_i^{yu}(z) & P_i^{y\nu}(z) \\ P_i^{\eta u}(z) & P_i^{\eta\nu}(z) \end{bmatrix} \begin{bmatrix} U_i(z) \\ \nu_i(z) \end{bmatrix}. \tag{10}$$

The vectors $\eta_i(k)$ and $\nu_i(k)$ are the stacking of the corresponding vectors indexed according to the out-neighbors and in-neighbors of node i, namely, $\eta_i(k) = \mathbf{vert}[\eta_{ji}(k)]_{j \in \mathcal{N}_i^+ \setminus \{i\}}$ and $\nu_i(k) = \mathbf{vert}[\nu_{ij}(k)]_{j \in \mathcal{N}_i^- \setminus \{i\}}$ $\forall\, i$. They correspond to the overall set of messages transmitted and received by P_i, where $\eta_{ji}(k)$ is the message vector transmitted from plant P_i to P_j at the time instant k and $\nu_{ij}(k)$ is the message received by P_i from P_j at time instant k.

We have the following assumption that holds throughout the paper.

Assumption 1: In this paper, we consider all the communication links of the network to be noiseless and delay-free.

Under the Assumption 1, the network interconnection equations can be written as

$$\nu_{ij}(k) = \eta_{ij}(k) \quad \forall j \in \mathcal{N}_i^- \setminus \{i\}, \forall, i \in \mathcal{V}. \tag{11}$$

1) Strictly Causal Interaction: While the graph's edges represent the physical communication links between sub-systems and thus describe the topology of the network, they do not completely describe the nature of interaction over the network. In particular, the setup so far allows the nodes to act in the network as instantaneous relays of their local inputs.

For example, in a strongly connected graph, this allows the possibility for any input $u_i(k)$ to instantaneously affect any other node even though there is no directed communication link from node i to node j. Similarly, each node would be able to receive the local measurements of any other node instantaneously. Thus, any strongly connected graph topology will effectively result in a all-to all communication topology. More generally, the graph and the communication topology will not coincide if instantaneous relays are allowed.[1]

We require instead the graph topology to reflect the communication topology even though the links are perfect. One way to obtain this is to not allow the nodes on the network to be instantaneous relays.

We complete the description of the networked systems considered in this paper with the following definition.

Definition 4: A networked system P is said to be a *strictly causal interaction of sub-systems over a given network* if the network outputs η_i of each sub-system P_i are only functions of their local state information, x_i ($D_{\eta_u} = 0$, $D_{\eta_\nu} = 0$) or equivalently $P_i^{\eta u}$ and $P_i^{\eta\nu}$ are strictly proper.

[1]We believe that this discrepancy has been a source of confusion in the literature.

Remark 2: Definition 5 induces a notion of spatial causality relative to the underlying graph. Each node's dynamics at time $k+1$ only depend on the in-neighbors' state variables at time k, and cannot depend on the two-step or farther neighbors' state variables at time k. Thus, the topology of the graph now has a stronger impact on the problem as it dictates the nature of interactions between the neighboring nodes, and coincides with the communication topology.

All the networked systems in this paper are obtained from strictly causal interactions of sub-systems over a given noiseless and delay-free network. This is the main difference between the system models considered in this paper and other works like [11], [17] where the nodes can act like instantaneous relays. The dynamic coupled systems used in [15], [16] can be viewed as a strictly causal interaction of sub-systems.

With abuse of terminology, we will often refer to strictly causal interactions of sub-systems over a delay-free and noiseless network simply as interconnected or networked systems.

Next, we consider the feedback representation for such systems.

B. Feedback Representation of Networked Systems

Remark 3: The representation (9) (10) and (11) is parsimonious as it involves the signals that are actually exchanged over the network. If a node i has no in-neighbors or out-neighbors, there are no input or output network signals associated with it. Thus, ν_i or η_i are not defined like the corresponding input columns or output rows in (9) and (10). To get a more compact representation of the networked system, it is convenient and without loss of generality to make the missing signals explicit and equal to zero. This is easily done by setting $\nu_i(k) = \nu_{i0}(k) = 0$ and $\eta_i(k) = \eta_{0i}(k) = 0$ and by padding (9) and (10) with a zero input column and zero output row when node i has no in-neighbors and out-neighbors, respectively. Whenever is convenient, like in this section, we assume that this padding has been done without changing the notation.

Based on Definition 4, define $\hat{P} = \mathbf{diag}[P_i]_i$ as a system with a state-space representation given by

$$\begin{bmatrix} x(k+1) \\ y(k) \\ \eta(k) \end{bmatrix} = \begin{bmatrix} \hat{A} & \hat{B}_u & \hat{B}_\nu \\ \hat{C}_y & \hat{D}_{yu} & \hat{D}_{y\nu} \\ \hat{C}_\eta & 0 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \\ \nu(k) \end{bmatrix} \tag{12}$$

where $\hat{A} = \mathbf{diag}[A_i]_i$, $\hat{B}_u = \mathbf{diag}[B_i^u]_i$, $\hat{B}_\nu = \mathbf{diag}[B_i^\nu]_i$, $\hat{C}_y = \mathbf{diag}[C_i^y]_i$, $\hat{D}_{yu} = \mathbf{diag}[D_i^{yu}]_i$, $\hat{D}_{y\nu} = \mathbf{diag}[D_i^{y\nu}]_i$, $\hat{C}_\eta = \mathbf{diag}[C_i^\eta]_i$.

The state, input and output vectors are given by $x(k) = \mathbf{vert}[x_i(k)]_i$, $u(k) = \mathbf{vert}[u_i(k)]_i$, $y(k) = \mathbf{vert}[y_i(k)]_i$, $\eta(k) = \mathbf{vert}[\eta_i(k)]_i$ and $\nu(k) = \mathbf{vert}[\nu_i(k)]_i$. Let the corresponding partitions of $x(k)$, $u(k)$, $y(k)$, $\eta(k)$, $\nu(k)$ be $\mathcal{P}_x$, $\mathcal{P}_u$, $\mathcal{P}_y$, $\mathcal{P}_\eta$ and $\mathcal{P}_\nu$, respectively. Based on these partitions, we can see that the matrices in (12) are partitioned as follows $\hat{A} - (\mathcal{P}_x, \mathcal{P}_x)$, $\hat{B}_u - (\mathcal{P}_x, \mathcal{P}_u)$, $\hat{B}_\nu - (\mathcal{P}_x, \mathcal{P}_\nu)$, $\hat{C}_y - (\mathcal{P}_y, \mathcal{P}_x)$, $\hat{C}_\eta - (\mathcal{P}_\eta, \mathcal{P}_x)$, $\hat{D}_{yu} - (\mathcal{P}_y, \mathcal{P}_u)$, $\hat{D}_{y\nu} - (\mathcal{P}_y, \mathcal{P}_\nu)$.

An equivalent formulation for $\hat{P}$ can be given using transfer function matrices

$$\hat{P}(z) = \begin{bmatrix} \hat{P}_{yu}(z) & \hat{P}_{y\nu}(z) \\ \hat{P}_{\eta u}(z) & \hat{P}_{\eta\nu}(z) \end{bmatrix} \tag{13}$$

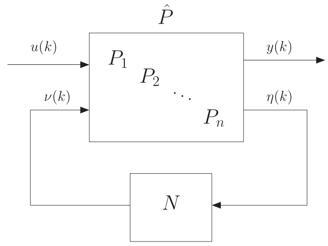

Fig. 2. A networked system expressed as a feedback interconnection of the sub-systems and the network interconnection matrix.

where $\hat{P}_{yu}(z) = \mathbf{diag}[P_i^{yu}(z)]_i$, $\hat{P}_{y\nu}(z) = \mathbf{diag}[P_i^{y\nu}(z)]_i$, $\hat{P}_{\eta u}(z) = \mathbf{diag}[P_i^{\eta u}(z)]_i$ and $\hat{P}_{\eta\nu}(z) = \mathbf{diag}[P_i^{\eta\nu}(z)]_i$, with $\hat{P}_{\eta u}(z)$ and $\hat{P}_{\eta\nu}(z)$ strictly proper.

It follows from (11) that

$$\nu(k) = N\eta(k) \tag{14}$$

for some matrix N, which is structured according to $\tilde{\mathcal{A}}(\mathcal{G})$ and partitioned according to $(\mathcal{P}_\nu, \mathcal{P}_\eta)$. We refer to N as *network interconnection matrix*. Note that the network interconnection matrix is a static gain matrix that can be obtained from the graph $\mathcal{G}$ and the dimensions of the message vectors $\nu_{ij}(k)$. An example of a system consistent with Fig. 1 is described in Section VII-A.

From the above notation, a networked system P is obtained by the feedback interconnection of $\hat{P}$ in (12) with N, as shown in Fig. 2, or $P = \mathrm{F}_l(\hat{P}, N)$, where F_l stands for lower Linear Fractional Transformation (LFT)[2] as defined in [24].

Definition 5: For given $\mathcal{G}$ and $\mathcal{P}_u, \mathcal{P}_y$, we denote by $\mathcal{N}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$ the set of networked systems $P = \mathrm{F}_l(\hat{P}, N)$ obtained by the strictly causal network interconnection of $\hat{P}$ defined in (12) and a network interconnection matrix N structured according to $\tilde{\mathcal{A}}(\mathcal{G})$ for appropriate $\mathcal{P}_x, \mathcal{P}_\eta, \mathcal{P}_\nu$.

We next analyze the resulting state-space and input-output structure of the networked systems in $\mathcal{N}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$.

C. State-Space and Input-Output Descriptions of Networked Systems

We can write the state-space representation of a networked system P based on (12) and (14) given by

$$\begin{aligned} P &= \mathrm{F}_l(\hat{P}, N) \\ &= \left[\begin{array}{c|c} \hat{A} + \hat{B}_\nu N \hat{C}_\eta & \hat{B}_u \\ \hline \hat{C}_y + \hat{D}_{y\nu} N \hat{C}_\eta & \hat{D}_{yu} \end{array}\right] := \left[\begin{array}{c|c} A & B_u \\ \hline C_y & D_{yu} \end{array}\right]. \end{aligned} \tag{15}$$

Lemma 4: Any networked system, P, that is a causal interaction over a delay-free and noiseless network $\mathcal{G}$ has a state-space representation that belongs to the set $\mathfrak{S}(\mathcal{G}, \mathcal{P}_x, \mathcal{P}_u, \mathcal{P}_y)$, for some state partition $\mathcal{P}_x$, and has a transfer function matrix $P(z)$ that belongs to $\mathfrak{T}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$.

Proof: Using the fact that N is structured according to $\tilde{\mathcal{A}}(\mathcal{G})$, partitioned according to $(\mathcal{P}_\nu, \mathcal{P}_\eta)$ and following the partitions of the state-space matrices in (15), Lemma 1 can be used to show that A, C_y are structured according to $\mathcal{A}(\mathcal{G})$ while B_u, D_{yu} are block-diagonal and the state-space matrices have consistent dimension and partitions.

From Lemma 3, it follows that transfer function matrix corresponding to any interconnected system that is a strictly causal interaction over a delay-free and noiseless network $\mathcal{G}$ belongs to $\mathfrak{T}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$. ■

IV. Implementing and Realizing Systems Over the Given Network

In the previous section, we looked at some of the properties of the state-space and input-output descriptions of interconnected systems that are strictly causal interactions of sub-systems over a given delay-free and noiseless network. It was shown that any such networked system has a state-space representation in $\mathfrak{S}(\mathcal{G}, \mathcal{P}_x, \mathcal{P}_u, \mathcal{P}_y)$ and an input-output representation in $\mathfrak{T}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$.

In this section, we address the reverse problem of expressing elements of $\mathfrak{S}(\mathcal{G}, \mathcal{P}_x, \mathcal{P}_u, \mathcal{P}_y)$ and $\mathfrak{T}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$ as strictly causal interactions of sub-systems over a given noiseless and delay-free network $\mathcal{G}$ with some extra requirement. Note that our definition allows for networked systems that are non-minimal. At the same time, from a practical implementation viewpoint, we are interested in networked systems that do not have hidden unstable modes. To address the problem, we introduce the notions of implementability and realizability of a system over a given network. These properties play an important role in the synthesis phase, since it is necessary to know if a designed system with a structure in $\mathfrak{S}(\mathcal{G}, \mathcal{P}_x, \mathcal{P}_u, \mathcal{P}_y)$ or $\mathfrak{T}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$ can be implemented as networked system over $\mathcal{G}$.

Definition 6: We call a system $P \in \mathfrak{S}(\mathcal{G}, \mathcal{P}_x, \mathcal{P}_u, \mathcal{P}_y)$ which is detectable and stabilizable and such that $P \in \mathcal{N}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$, *network implementable* and denote the set of all such systems by $\mathcal{N}^I(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$.

In other words, a system $P \in \mathfrak{S}(\mathcal{G}, \mathcal{P}_x, \mathcal{P}_u, \mathcal{P}_y)$ is said to be implementable over a given network $\mathcal{G}$ if it is stabilizable, detectable and can be expressed as a strictly causal interaction of some sub-systems over the given network. Definition 6 excludes non stabilizable and/or non detectable systems from those we consider implementable over the network. This exclusion is important but natural since an interconnected system that is not stabilizable or detectable has hidden unstable modes not visible from the input-output map. These modes will make the networked system useless in practice as noise or uncertainty, always present in the network interconnections, will excite them and manifest the "internal" instability. Thus, although we assume ideal network interconnections in this paper, we do not want our results to be fragile to this assumption.

For example, the following third-order state-space realization is in $\mathfrak{S}(\mathcal{G}, \mathcal{P}_x, \mathcal{P}_u, \mathcal{P}_y)$, with $\mathcal{G} = \{1 \to 3 \to 2 \to 1\}$ and $\mathcal{P}_x = \mathcal{P}_y = \mathcal{P}_u = [1, 1, 1]$

$$C = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix}, A = \begin{bmatrix} 3 & -1 & 0 \\ 0 & 3 & -1 \\ -1 & 0 & 3 \end{bmatrix}, B = I,\ D = 0. \tag{16}$$

The eigenvalues of A are at $-3.5 \pm i(\sqrt{3}/2)$ and at 2. This last unstable mode is not observable. Because of the lack of detectability, this system is not network implementable according to Definition 6, and is of little practical use.

Definition 7: A system P with transfer function matrix $P(z) \in \mathfrak{T}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$ is said to be *realizable over* $\mathcal{G}$ if there

[2]Note that having $D_i^{\eta u} = 0$, and $D_i^{\eta\nu} = 0\,\forall i$ ensures that the feedback interconnection of $\hat{P}$ and N is well-posed.

exists a system $\tilde{P}$ with partitions $\mathcal{P}_u$, $\mathcal{P}_y$ *network implementable* over $\mathcal{G}$ (for some $\mathcal{P}_x$), i.e., $\tilde{P} \in \mathcal{N}^I(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$, such that $P(z) = \mathbf{tf}(\tilde{P})$ according to (5).

Note that the definition of network realizability does not impose a condition for the state-space realization of a transfer function to be minimal. Instead, the state-space realization needs to be stabilizable and detectable in order to be realizable over the network.

Given a transfer function matrix in $\mathfrak{T}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$, we are interested to know if the system is network realizable and, if it is, we want to know how to realize it.

We observe that not all transfer function matrices in $\mathfrak{T}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$ are network realizable, [22]. We leave it to the reader to verify that $P(z)$ in (7) is not, since any realization in $\mathfrak{S}(\mathcal{G}, \mathcal{P}_x, \mathcal{P}_u, \mathcal{P}_y)$ will require one to repeat the unstable mode at 2, based on the argument in Remark 1. This mode, invisible in the impulse response, makes the system not stabilizable and/or detectable. However, checking the network realizability of a system is not easy in general.

Note also that following the definition by searching for structured A, B, C, D to match the impulse response is not practical, although it can be insightful in special cases. For example, consider $P(z)$ in (8). When, we search for a third-order structured system $\tilde{P}$ ($\tilde{P} \in \mathfrak{S}(\mathcal{G}, \mathcal{P}_x, \mathcal{P}_u, \mathcal{P}_y)$ with $\mathcal{P}_x = [1, 1, 1]$) so that $P(z) = \mathbf{tf}(\tilde{P})$, we obtain (16), which is not a network realization of (8) as it is non detectable. In fact, this structured realization is unique, modulo scaling of the diagonal elements of B and correspondingly of the columns of C. Thus, the transfer function (8) is not network realizable as a third order system. However, this does not exclude the possibility of the system being network realizable for a different $\mathcal{P}_x$.

A. Network Implementability

From the previous development, we know that if $P \in \mathcal{N}^I(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$ then $P = \mathrm{F}_l(\hat{P}, N) \in \mathfrak{S}(\mathcal{G}, \mathcal{P}_x, \mathcal{P}_u, \mathcal{P}_y)$, for some $\mathcal{P}_x$ and some $\hat{P}$, and it is detectable and stabilizable. The main contribution of this section is to show that every stabilizable and detectable element of the set $\mathfrak{S}(\mathcal{G}, \mathcal{P}_x, \mathcal{P}_u, \mathcal{P}_y)$ is implementable over the given network $\mathcal{G}$. This result allows us to characterize all the implementable systems on a given network in terms of their state-space description.

Lemma 5: Any stabilizable and detectable system $Q \in \mathfrak{S}(\mathcal{G}, \mathcal{P}_x, \mathcal{P}_u, \mathcal{P}_y)$ is *network implementable* (cf. Def. 6) over $\mathcal{G}$, i.e., $Q \in \mathcal{N}^I(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$.

Proof: By hypothesis, Q is stabilizable and detectable. To prove the lemma, we need to find n sub-systems $\{Q_i\}_i$ with state-space representation of the form (9) which result in the given Q by interconnecting them over the delay-free and noiseless network $\mathcal{G}$ while maintaining stabilizability and detectability.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. By definition, Q has a state-space realization (A, B_u, C_y, D_{yu}), where A, C_y are structured according to $\mathcal{A}(\mathcal{G})$ while B_u, D_{yu} are block-diagonal and the state-space matrices have consistent dimensions and partitions.

$$\begin{aligned} x_i(k+1) &= \sum_{j \in \mathcal{N}_i^-} A_{ij} x_j(k) + B_i^u u_i(k) \\ y_i(k) &= \sum_{j \in \mathcal{N}_i^-} C_{ij}^y x_j(k) + D_i^{yu} u_i(k) \quad \forall i \in \mathcal{V}. \end{aligned} \tag{17}$$

Define n sub-systems $\{Q_i\}_i$ given by

$$Q_i : \begin{bmatrix} x_i(k+1) \\ y_i(k) \\ \eta_i(k) \end{bmatrix} = \begin{bmatrix} A_i & B_i^u & B_i^\nu \\ C_i^y & D_i^{yu} & D_i^{y\nu} \\ C_i^\eta & 0 & 0 \end{bmatrix} \begin{bmatrix} x_i(k) \\ u_i(k) \\ \nu_i(k) \end{bmatrix} \quad \forall i \in \mathcal{V}$$

where

$$\begin{aligned} &A_i = A_{ii},\ C_i^y = C_{ii}^y,\ B_i^\nu = \mathbf{hor}[A_{ij}]_{j \in \mathcal{N}_i^- \setminus \{i\}} \\ &D_i^{y\nu} = \mathbf{hor}[C_{ij}]_{j \in \mathcal{N}_i^- \setminus \{i\}},\ C_i^\eta = \mathbf{vert}[I]_{j \in \mathcal{N}_i^+ \setminus \{i\}} \quad \forall i \in \mathcal{V}. \end{aligned} \tag{18}$$

Note that (18) leads to $\eta_{ji}(k) = x_i(k)$ for all $j \in \mathcal{N}_i^+ \setminus \{i\}$ and $i \in \mathcal{V}$, which implies that

$$\eta_i(k) = \mathbf{vert}\left[\eta_{ji}(k)\right]_{j \in \mathcal{N}_i^+ \setminus \{i\}} = \mathbf{vert}\left[x_i(k)\right]_{j \in \mathcal{N}_i^+ \setminus \{i\}}. \tag{19}$$

Now, define $\hat{Q} = \mathbf{diag}[Q_i]_i$ according to (12) and let the network interconnection equations $\nu_{ij}(k) = \eta_{ij}(k)$ for all $j \in \mathcal{N}_i^- \setminus \{i\}, i \in \mathcal{V}$ be expressed by (14) using a network interconnection matrix N. Following (18) and applying the zero padding of Remark 3, it is easy to show that $\mathrm{F}_l(\hat{Q}, N)$ is the same as the given system Q. Also note that N is a static gain matrix and the number of states in $\hat{Q}$ are the same as the number of states in Q. So, no hidden unstable modes are introduced and thus the stabilizability and detectability properties are preserved during the implementation of Q as strictly causal interactions of sub-systems over the given network $\mathcal{G}$.

Thus, every stabilizable and detectable $Q \in \mathfrak{S}(\mathcal{G}, \mathcal{P}_x, \mathcal{P}_u, \mathcal{P}_y)$ is implementable over the network $\mathcal{G}$. ■

Remark 4: The reader should note that the constructive proof works also without stabilizability and detectability assumption. Thus, any element in $\mathfrak{S}(\mathcal{G}, \mathcal{P}_x, \mathcal{P}_u, \mathcal{P}_y)$ corresponds to a networked system composed of strictly causal interactions of sub-systems over a given delay-free and noiseless network. However, we are only interested in those that are stabilizable and detectable, because they can be physically built over the network (i.e., without internal instabilities).

Definition 8: Given a network $\mathcal{G}$, the set of all the systems that are strictly causal interactions of sub-systems over the noiseless and delay-free network $\mathcal{G}$ with input and output partitions $\mathcal{P}_u$ and $\mathcal{P}_y$, respectively, is denoted by $\mathfrak{S}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y) = \bigcup_{\mathcal{P}_x \in \mathbb{N}^n} \mathfrak{S}(\mathcal{G}, \mathcal{P}_x, \mathcal{P}_u, \mathcal{P}_y)$.

Since any implementable system on the network $\mathcal{G}$, with input and output partitions as $\mathcal{P}_u$ and $\mathcal{P}_y$, is a stabilizable and detectable element of $\mathfrak{S}(\mathcal{G}, \mathcal{P}_x, \mathcal{P}_u, \mathcal{P}_y)$ for some state partition $\mathcal{P}_x$, and any stabilizable and detectable element of $\mathfrak{S}(\mathcal{G}, \mathcal{P}_x, \mathcal{P}_u, \mathcal{P}_y)$ is implementable over $\mathcal{G}$ for any $\mathcal{P}_x$, we can say that the set of stabilizable and detectable elements in $\mathfrak{S}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$ represents the set of all discrete-time, causal FDLTI systems that are implementable over a noiseless and delay-free network $\mathcal{G}$. For future reference, we denote the set of all stable systems in $\mathfrak{S}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$ by $\mathfrak{S}^s(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$. Note that all the elements of $\mathfrak{S}^s(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$ are implementable over $\mathcal{G}$.

B. Network Realizability

We now turn our attention to network realizability, which is more stringent than network implementability by definition. In general, we do not yet know simple conditions for checking whether a given $P(z) \in \mathfrak{T}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$ is network realizable, or general realization procedures in case the system is known to be network realizable. However, in this section, we show that any stable system in $\mathfrak{T}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$ is realizable over the network $\mathcal{G}$ by providing a realization method. The network realizability of stable systems plays an important role in the parametrization of all stabilizing network realizable controllers for networked plants.

Theorem 1: Given a network represented by a directed pseudograph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and the input and output partitions, $\mathcal{P}_u$ and $\mathcal{P}_y$, any bounded-input bounded-output (BIBO) stable system $Q(z) \in \mathfrak{T}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$ is realizable over the given network.

Proof: The reader should refer to the Appendix for the proof and a numerical example. ■

Remark 5: When applied to unstable systems, the constructive procedure in the proof of Theorem 1, which is based on repeating poles, produces non-minimal realizations in $\mathfrak{S}(\mathcal{G}, \mathcal{P}_x, \mathcal{P}_u, \mathcal{P}_y)$, which are not stabilizable and/or detectable, and thus not network implementable over $\mathcal{G}$ by Definition 6. Therefore, when applied to unstable systems, the procedure cannot be used to draw conclusions regarding the network realizability of the system.

The underlying focus of the rest of the paper is to find conditions and other procedures to obtain a network implementable realization even for unstable systems (controllers) in $\mathfrak{T}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$.

For future reference, we denote the set of all stable real-rational proper transfer function matrices in $\mathfrak{T}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$ by $\mathfrak{T}^s(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$. Note that, if $Q(z) = [Q_{ij}(z)]_{i,j} \in \mathfrak{T}^s(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$, then $Q_{ij}(z) \in \mathcal{RH}_\infty$ for all i, j.

V. Feedback Interconnection of Networked Plant and Network Realizable Controller

So far, we have introduced a class of networked systems, analyzed their structures and properties, and posed and addressed the question of network implementability and realizability for given systems. In this section, we naturally extend the networked systems to include exogenous and control inputs as well as regulated and measured outputs, and consider the problem of feedback stabilization of a networked system by a network implementable controller.

A. Networked Plant Model

A networked plant P is defined as in (12), but with each sub-system now including local exogenous inputs $w_i(k)$ and local regulated outputs $z_i(k)$. Thus, the state-space description of the sub-system P_i is written as

$$P_i : \begin{bmatrix} x_i(k+1) \\ z_i(k) \\ y_i(k) \\ \eta_i(k) \end{bmatrix} = \begin{bmatrix} A_i & B_i^w & B_i^u & B_i^\nu \\ C_i^z & D_i^{zw} & D_i^{zu} & D_i^{z\nu} \\ C_i^y & D_i^{yw} & D_i^{yu} & D_i^{y\nu} \\ C_i^\eta & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_i(k) \\ w_i(k) \\ u_i(k) \\ \nu_i(k) \end{bmatrix} \quad (20)$$

where B_i^w, C_i^z, D_i^{zw}, D_i^{yw}, D_i^{zu} and $D_i^{z\nu}$ have dimensions compatible with the local exogenous inputs $w_i(k)$, local regulated outputs $z_i(k)$, local control inputs $u_i(k)$, the local measurement outputs $y_i(k)$, the local network outputs $\eta_i(k)$ and the local network inputs $\nu_i(k)$.

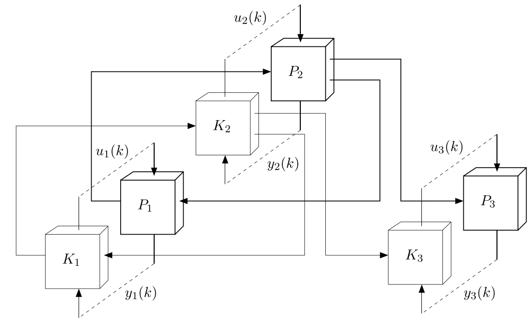

Fig. 3. An example of an interconnected plant and controller pair that are realizable over the network given in Fig. 1(b).

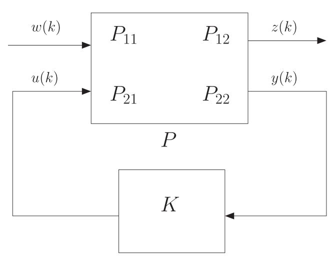

Fig. 4. Feedback interconnection of a plant and a controller.

A state-space representation for the networked plant $P = \mathrm{F}_l(\hat{P}, N)$ is given by the following expression:

$$P = \left[\begin{array}{c|c|c} \hat{A} + \hat{B}_\nu N \hat{C}_\eta & \hat{B}_w & \hat{B}_u \\ \hline \hat{C}_z + \hat{D}_{z\nu} N \hat{C}_\eta & \hat{D}_{zw} & \hat{D}_{zu} \\ \hline \hat{C}_y + \hat{D}_{y\nu} N \hat{C}_\eta & \hat{D}_{yw} & \hat{D}_{yu} \end{array}\right] := \begin{bmatrix} A & B_w & B_u \\ C_z & D_{zw} & D_{zu} \\ C_y & D_{yw} & D_{yu} \end{bmatrix}. \quad (21)$$

Note that $A := [A_{ij}]_{i,j}$, $C_z := [C_{ij}^z]_{i,j}$, $C_y := [C_{ij}^y]_{i,j}$ are structured according to $\mathcal{A}(\mathcal{G})$, while $B_w = \mathbf{diag}[B_i^w]_i$, $B_u = \mathbf{diag}[B_i^u]_i$, $D_{zw} = \mathbf{diag}[D_i^{zw}]_i$, $D_{zu} = \mathbf{diag}[D_i^{zu}]_i$, $D_{yw} = \mathbf{diag}[D_i^{yw}]_i$, $D_{yu} = \mathbf{diag}[D_i^{yu}]_i$ have a block diagonal structure.

One can view the system P as shown in Fig. 4 while the actual plant or process is in fact P_{22}, which is the map from $u(k)$ to $y(k)$. Note that, $P_{22} = (A, B_u, C_y, D_{yu}) \in \mathfrak{S}(\mathcal{G}, \mathcal{P}_x, \mathcal{P}_u, \mathcal{P}_y)$ where $\mathcal{P}_x$, $\mathcal{P}_u$ and $\mathcal{P}_y$ are the partitions of $x(k)$, $u(k)$ and $y(k)$, respectively, while A and C_y are structured according to $\mathcal{A}(\mathcal{G})$ and B_u and D_{yu} are block-diagonal.

B. Feedback Interconnection of Networked Plant and a Stabilizing Networked Controller

Next we look at the feedback interconnection of a networked plant P in (21), assumed to be detectable from y and stabilizable from u, and a stabilizing networked controller $K \in \mathfrak{S}(\mathcal{G}, \mathcal{P}_x^K, \mathcal{P}_y, \mathcal{P}_u)$ (for some partition $\mathcal{P}_x^K$) as shown in Fig. 3 and compactly in Fig. 4.

The state-space equations corresponding to the sub-systems of P are given by (21). For any controller $K = (A_K, B_K, C_K,$

$D_K) \in \mathfrak{S}(\mathcal{G}, \mathcal{P}_x^K, \mathcal{P}_y, \mathcal{P}_u)$, the state-space equations are given by

$$K: \begin{bmatrix} x^K(k+1) \\ u(k) \end{bmatrix} = \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix} \begin{bmatrix} x^K(k) \\ y(k) \end{bmatrix} \quad (22)$$

where the matrices A_K and C_K are structured according to $\mathcal{A}(\mathcal{G})$, while B_K and D_K are block diagonal.

Let the closed-loop system formed by the feedback interconnection of P and K be denoted by $T_{zw} = \mathrm{F}_l(P, K)$. Then combining (21) and (22) leads to the following state-space representation for the closed-loop system $T_{zw} := (A_C, B_C, C_C, D_C) =$ (23), as shown at the bottom of the page, where $M := (I - D_K D_{yu})^{-1}$ and $\hat{M} := (I + D_{yu} M D_K)$, which are assumed to exist (well-posed feedback interconnection), are block-diagonal and partitioned according to $(\mathcal{P}_u, \mathcal{P}_u)$ and $(\mathcal{P}_y, \mathcal{P}_y)$, respectively. From the structure of (23), we can see that the matrices in A_C and C_C are structured according to $\mathcal{A}(\mathcal{G})$ while those in B_C and D_C are block-diagonal. Using a simple rearrangement of the states of T_{zw}, we can show that there exists an equivalent state-space representation, to that of (23), in $\mathfrak{S}(\mathcal{G}, \mathcal{P}_x + \mathcal{P}_x^K, \mathcal{P}_w, \mathcal{P}_z)$.

Definition 9: $K = (A_K, B_K, C_K, D_K) \in \mathfrak{S}(\mathcal{G}, \mathcal{P}_x^K, \mathcal{P}_y, \mathcal{P}_u)$ is a stabilizing controller of P in (21), if A_C in (23) has all its eigenvalues strictly inside the unit disc, i.e., if T_{zw} in (23) is stable.

If K is a stabilizing controller, it is detectable from its output u and stabilizable from its input y. From Lemma 5, it can be seen that K is network implementable over $\mathcal{G}$. Note that the standard notion of closed-loop stability of Definition 9 and the result of Lemma 5 imply the BIBO stability of the closed loop networked system from any input, including those entering the communication links, to any output including the network signal, (details omitted for brevity). Moreover, if K is stabilizing, T_{zw} has a state-space representation in $\mathfrak{S}^s(\mathcal{G}, \mathcal{P}_x + \mathcal{P}_x^K, \mathcal{P}_w, \mathcal{P}_z)$. Following Lemma 5, it can be seen that the feedback interconnection of the considered P and K is also a strictly causal interaction of sub-systems over the given delay-free and noiseless network.

VI. All Stabilizing Network Implementable Controllers

In this section, we show that if a given networked plant satisfies a certain condition, we can parametrize the set of all stabilizing networked controllers implementable over the associated network.

Theorem 2: Given networked plant P over a noiseless and delay-free network $\mathcal{G}$ detectable from y and stabilizable from u. If there exist matrices F and L such that $A + B_u F$ and $A + LC_y$ are stable, where F is structured according to $\mathcal{A}(\mathcal{G})$ partitioned according to $(\mathcal{P}_u, \mathcal{P}_x)$ and L block-diagonal partitioned according to $(\mathcal{P}_x, \mathcal{P}_y)$. Then the set of all stabilizing FDLTI-DT controllers for P that are implementable over $\mathcal{G}$ is parametrized by

$$K = \mathrm{F}_l(J, Q) \quad (24)$$

where

$$J = \left[\begin{array}{c|cc} A + B_u F + LC_y + LD_{yu}F & -L & (B_u + LD_{yu}) \\ \hline F & 0 & I \\ -(C_y + D_{yu}F) & I & -D_{yu} \end{array}\right] := \left[\begin{array}{c|cc} A_J & -L & B_J \\ \hline F & 0 & I \\ C_J & I & D_J \end{array}\right] \quad (25)$$

is implementable over the network $\mathcal{G}$, Q is any FDLTI-DT system, stable and implementable over the network $\mathcal{G}$, and $I + Q(\infty)D_{yu}$ is nonsingular.

Before providing the proof, we would like to clarify its main point. For a general plant, the set of all stabilizing controllers is constructed from a model based controller and any free stable system Q. In our case, the plant P_{22} is networked and implementable. The model based controller is thus also networked, i.e., is implementable over the network, provided that suitable F and L can be found. The set of stabilizing controllers for P_{22} would then be obtained by selecting stable systems Q, but in principle without any network structure. Theorem 2 shows that Q can be restricted to stable networked realizable systems.

Proof: We use the formulation given in [24] to obtain all stabilizing controllers implementable over the network $\mathcal{G}$. It is well-known that all stabilizing discrete-time causal and FDLTI controllers are given by $K = F_l(J, Q)$, where J is given by (25) and Q is FDLTI, causal and stable. We include the property of network implementability to the above mentioned properties and prove the theorem in two steps.

First, assume that Q is stable and implementable over $\mathcal{G}$, i.e., $Q \in \mathfrak{S}^s(\mathcal{G}, \mathcal{P}_x^Q, \mathcal{P}_y, \mathcal{P}_u)$ for some state partition $\mathcal{P}_x^Q$. Let $Q = (A_Q, B_Q, C_Q, D_Q)$ where A_Q and C_Q are structured according to $\mathcal{A}(\mathcal{G})$ while B_Q and D_Q are block diagonal and all matrices are partitioned accordingly. Note that A_Q has its eigenvalues strictly inside the unit disc. Let $M_K = (I - D_Q D_J)^{-1}$ and $\hat{M}_K = (I + D_J M_K D_Q)$. Using the state-space star product formula for calculating the LFT of J and Q, we obtain the state-space matrices for K as (26), as shown at the bottom of the next page. Since A, C_y, F, A_Q and C_Q are structured according to $\mathcal{A}(\mathcal{G})$ while B_u, L, B_Q, and D_Q are block-diagonal, each of the four blocks of A_K and the two blocks of C_K are structured according to $\mathcal{A}(\mathcal{G})$. Also the two blocks of B_K and D_K are block diagonal. It is not difficult to verify that $K \in \mathfrak{S}(\mathcal{G}, \mathcal{P}_x + \mathcal{P}_x^Q, \mathcal{P}_y, \mathcal{P}_u)$, by regrouping the states of K as $\mathbf{vert}\begin{bmatrix} x_i^J(k) \\ x_i^Q(k) \end{bmatrix}_i$, where $x_J(k) = \mathbf{vert}[x_i^J(k)]_i$ is partitioned according to P_x and $x_Q(k) = \mathbf{vert}[x_i^Q(k)]_i$ is partitioned according to P_x^Q. We leave the details to the reader.

$$\left[\begin{array}{cc|c} A + B_u M D_K C_y & B_u M C_K & B_w + B_u M D_K D_{yw} \\ B_K \hat{M} C_y & A_K + B_K D_{yu} M C_K & B_K \hat{M} D_{yw} \\ \hline C_z + D_{zu} M D_K C_y & D_{zu} M C_K & D_{zw} + D_{zu} M D_K D_{yw} \end{array}\right] \quad (23)$$

From the theory of Youla parameterization, we know that K given by (26) is a stabilizing controller for the given P. Thus, K is also stabilizable and detectable. Using Lemma 5, we say that K is implementable over the network $\mathcal{G}$.

Next, we prove that if K is implementable over $\mathcal{G}$ then Q is also implementable over $\mathcal{G}$. Note that K is stabilizing, FDLTI and causal, while Q is stable, FDLTI and causal. From standard results on LFT, we know that $Q = \mathrm{F}_l(\hat{J}, K)$ where

$$\hat{J} = \left[\begin{array}{c|cc} A & -L & B_u \\ \hline -F & 0 & I \\ C_y & I & D_{yu} \end{array}\right] \tag{27}$$

and $(I - D_K D_{yu})$ is invertible. Following a similar procedure as before, we can see that $Q \in \mathfrak{S}(\mathcal{G}, \mathcal{P}_y, \mathcal{P}_u)$ and in particular $Q \in \mathfrak{S}^s(\mathcal{G}, \mathcal{P}_y, \mathcal{P}_u)$. Since Q is stable, it is stabilizable and detectable. Thus Q is implementable over $\mathcal{G}$ if K is implementable over $\mathcal{G}$. ■

Remark 6: For the large class of networked systems considered in this paper, which are strictly causal interactions of sub-systems over a given network, Theorem 2 together with Theorem 1 provide a constructive procedure to obtain stabilizing networked controllers in terms of sub-controllers interacting over a the same network as the plant.

A. *Sufficiency Conditions for Constructing F and L*

Theorem 2 requires a matrix F structured according to $\mathcal{A}(\mathcal{G})$ and partitioned according to $(\mathcal{P}_u, \mathcal{P}_x)$ such that $A + B_u F$ is Schur-stable. Similarly, it also requires a block-diagonal matrix L partitioned according to $(\mathcal{P}_x, \mathcal{P}_y)$ such that $A + LC_y$ is Schur-stable. The theorem provides a characterization of all-stabilizing controllers implementable over the given network based on the matrices F and L satisfying the above mentioned constraints. In this section, we provide constructive algorithms to obtain such matrices F and L. *Note that for stable systems, F and L can always be chosen to be zero matrices*.

The stability test for discrete-time systems is given by a discrete-time Lyapunov equation. In [25], the stability test has been expressed as a feasibility problem as shown in the following lemma.

Lemma 6: A matrix A is Schur-stable if, and only if, there exist a symmetric matrix $M = M'$ and a general matrix G such that the LMI

$$\begin{bmatrix} M & AG \\ G'A' & G + G' - M \end{bmatrix} \succ 0 \tag{28}$$

is feasible.

We extend Lemma 6 to construct matrices F and L with the previously mentioned properties by solving a convex feasibility problem.

Lemma 7: Given matrices A and B_u that are partitioned according to $(\mathcal{P}_x, \mathcal{P}_x)$ and $(\mathcal{P}_x, \mathcal{P}_u)$, respectively, there exists a matrix F that is structured according to $\mathcal{A}(\mathcal{G})$ and partitioned according to $(\mathcal{P}_u, \mathcal{P}_x)$ such that $A + B_u F$ is Schur-stable if there exist a symmetric matrix $M = M'$ and a matrix G block-diagonal and partitioned according to $(\mathcal{P}_x, \mathcal{P}_x)$ and a matrix R structured according to $\mathcal{A}(\mathcal{G})$ and partitioned according to $(\mathcal{P}_u, \mathcal{P}_x)$ such that the following LMI:

$$\begin{bmatrix} M & AG + B_u R \\ (AG + B_u R)' & G + G' - M \end{bmatrix} \succ 0 \tag{29}$$

is feasible. Moreover $F = RG^{-1}$.

Proof: If (29) has a solution, then $G + G' \succ M \succ 0$ which implies that G is non-singular and thus G^{-1} exists. Combining (29) with Lemma 6, we note that $A + B_u RG^{-1}$ is Schur-stable. Due to the structure of R and G, it is easy to see that $F = RG^{-1}$ is a matrix that is structured according to $\mathcal{A}(\mathcal{G})$ and partitioned according to $(\mathcal{P}_u, \mathcal{P}_x)$ and $A + B_u F$ is Schur-stable. ■

Lemma 8: Given matrices A and C_y that are partitioned according to $(\mathcal{P}_x, \mathcal{P}_x)$ and $(\mathcal{P}_y, \mathcal{P}_x)$, respectively, there exists a block-diagonal matrix L that is partitioned according to $(\mathcal{P}_x, \mathcal{P}_y)$ such that $A + LC_y$ is Schur-stable if there exist a symmetric matrix $M = M'$, a matrix G block-diagonal partitioned according to $(\mathcal{P}_x, \mathcal{P}_x)$ and a block-diagonal matrix R partitioned according to $(\mathcal{P}_y, \mathcal{P}_x)$ such that the following LMI:

$$\begin{bmatrix} M & A'G + C_y' R \\ G'A + R'C_y & G + G' - M \end{bmatrix} \succ 0 \tag{30}$$

is feasible. Moreover $L = (RG^{-1})'$.

Proof: The proof is similar to that of Lemma 7. ■

The above conditions appear to be new. Their conservatism may be reduced by allowing G to have appropriate structure compatible with A, we leave this extension to further investigations. For an alternative approach to finding F and L see [26]. We leave other ways of deriving network implementable central controller J to future investigations.

VII. Optimal Solution for the Distributed $\mathcal{H}_2$ Problem

We are now able to present the final contribution of this paper. In this section, under the assumptions of Theorem 2, we provide an optimal network implementable controller for the distributed $\mathcal{H}_2$ control problem. Fig. 3 depicts such a plant-controller pair when both are constrained to be interconnected systems over the same network $\mathcal{G}$.

As we now know, a distributed controller implementable over the network $\mathcal{G}$ can be seen as a stabilizable and detectable system in $\mathfrak{S}(\mathcal{G}, \mathcal{P}_y, \mathcal{P}_u)$.[3] Thus, given a plant P in (21) stabilizable from u and detectable from y implementable over a network $\mathcal{G}$

[3]Note that the input and output partitions of the controller have to match the output and input partitions of the interconnected plant, while the state partition is not fixed.

$$\left[\begin{array}{cc|c} A_J + B_J M_K D_Q C_J & B_J M_K C_Q & -L + B_J M_K D_Q \\ B_Q \hat{M}_K C_J & A_Q + B_Q D_J M_K C_Q & B_Q \hat{M}_K \\ \hline F + M_K D_Q C_J & M_K C_Q & M_K D_Q \end{array}\right] := \left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & D_K \end{array}\right] \tag{26}$$

and the assumptions of Theorem 2 hold, the network distributed $\mathcal{H}_2$ control problem can be written as

$$\begin{aligned} \min \quad & \|T_{zw}\|_2 \\ \text{subject to} \quad & K \in \mathfrak{S}(\mathcal{G}, \mathcal{P}_y, \mathcal{P}_u) \\ & K \text{ stabilizing} \end{aligned} \tag{31}$$

where $T_{zw} = \mathrm{F}_l(P, K)$ denotes the closed-loop mapping from $w(k)$ to $z(k)$. Then, from Theorem 2

$$T_{zw} = F_l\left(P, F_l(J, Q)\right)$$

where J is given by (25) and $Q \in \mathfrak{S}^s(\mathcal{G}, \mathcal{P}_y, \mathcal{P}_u)$. If there exists matrices F and L with the properties described in the hypothesis of Theorem 2, the set of all closed-loop transfer matrices from $w(k)$ to $z(k)$ by an internally stabilizing proper controller implementable over the network $\mathcal{G}$ can be obtained using Theorem 2 and the results from [24] as

$$\begin{aligned} T_{zw} = \mathrm{F}_l(T, Q) = \{T_{11} + T_{12} Q T_{21} :\ & Q \in \mathfrak{S}^s(\mathcal{G}, \mathcal{P}_y, \mathcal{P}_u), \\ & I + Q(\infty) D_{yu} \text{ invertible}\} \end{aligned} \tag{32}$$

where T is given by

$$\begin{aligned} T &= \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \\ &= \left[\begin{array}{cc|cc} A + B_u F & -B_u F & B_w & B_u \\ 0 & A + LC_y & B_w + LD_{yw} & 0 \\ \hline C_z + D_{zu} F & -D_{zu} F & D_{zw} & D_{zu} \\ 0 & C_y & D_{yw} & 0 \end{array}\right]. \end{aligned} \tag{33}$$

Since the closed-loop transfer matrix is simply an affine function of the controller parameter matrix Q, we can rewrite the distributed $\mathcal{H}_2$ problem in (31) as a convex optimization problem

$$\begin{aligned} \min \quad & \|T_{11} + T_{12} Q T_{21}\|_2 \\ \text{subject to} \quad & Q \in \mathfrak{S}^s(\mathcal{G}, \mathcal{P}_y, \mathcal{P}_u) \end{aligned} \tag{34}$$

It is convenient to solve the above problem in the frequency domain. This is possible since $Q \in \mathfrak{S}^s(\mathcal{G}, \mathcal{P}_y, \mathcal{P}_u)$ is equivalent to $Q(z) \in \mathfrak{T}^s(\mathcal{G}, \mathcal{P}_y, \mathcal{P}_u)$. We can maintain the implementability of the controller from the results of Theorem 1, which guarantees the realizability of $Q(z)$ with a state-space realization $\tilde{Q} \in \mathfrak{S}^s(\mathcal{G}, \mathcal{P}_y, \mathcal{P}_u)$ implementable over the network $\mathcal{G}$. Thus, the optimization problem in (34) can equivalently be expressed as

$$\begin{aligned} \min \quad & \|T_{11}(z) + T_{12}(z) Q(z) T_{21}(z)\|_2 \\ \text{subject to} \quad & Q(z) \in \mathfrak{T}^s(\mathcal{G}, \mathcal{P}_y, \mathcal{P}_u). \end{aligned} \tag{35}$$

The problem is now reduced to a standard convex optimization form, where $Q(z)$ is structured ([9], [12]). However, once the optimal $Q(z)$ is found, in terms of the impulse response or a state-space realization, we would not obtain the optimal controller as $K = F_l(J, Q(z))$. K would have the right input-output structure, i.e., $K \in \mathfrak{T}(\mathcal{G}, \mathcal{P}_y, \mathcal{P}_u)$, but its state-space structure will be destroyed i.e., $K \notin \mathfrak{S}(\mathcal{G}, \mathcal{P}_y, \mathcal{P}_u)$. Thus, the solutions have the right input-output structure, they can be implemented in a centralized way (cf. Section II-C), but their networked realization is generally not known. Instead we obtain a network implementable state-space realization $\tilde{Q} \in \mathfrak{S}^s(\mathcal{G}, \mathcal{P}_y, \mathcal{P}_u)$ for $Q(z)$ using Theorem 1, and then a network implementable state-space realization for the optimal K from Theorem 2, even if K is unstable.

The optimization problems like (35) to be computationally solved are transformed into vector problems [27]. Namely, by rewriting $T_{11}(z) + T_{12}(z) Q(z) T_{21}(z)$ as $h(z) - \mathcal{A}(z) q(z)$ where $h(z)$ and $q(z)$ are transfer function vectors of appropriate dimensions. Here we adapt the technique used in [12] to our notation to rewrite the optimization problem in (35) as an equivalent unconstrained problem. To represent the vectorization of a transfer function matrix, we make a slight change of notation for representing the matrices. Instead of treating $Q_{ij}(z)$ as a sub-matrix of $Q(z)$, we consider $Q_{ij}(z)$ to be the element of the matrix $Q(z)$ in the ith row and jth column.

Let $\mathbf{vec}(\mathfrak{T}^s(\mathcal{G}, \mathcal{P}_y, \mathcal{P}_u)) = \{\mathbf{vec}(Q(z)) | Q(z) \in \mathfrak{T}^s(\mathcal{G}, \mathcal{P}_y, \mathcal{P}_u)\}$ denote the set of vectorized elements of $\mathfrak{T}^s(\mathcal{G}, \mathcal{P}_y, \mathcal{P}_u)$. If $\mathcal{P}_u = \{\mathcal{P}_u^1, \ldots, \mathcal{P}_u^n\}$ denotes the output partition, then denote $n_u = \sum_i \mathcal{P}_u^i$ to represent the total number of outputs. Similarly, denote n_y to represent the total number of inputs. It can be seen that $\mathbf{vec}(\mathfrak{T}^s(\mathcal{G}, \mathcal{P}_y, \mathcal{P}_u)) \in \mathcal{RH}_\infty^{n_u n_y \times 1}$ is a sub-space due to the delay and sparsity constraints imposed by the set $\mathfrak{T}^s(\mathcal{G}, \mathcal{P}_y, \mathcal{P}_u)$. Let a denote the total number of elements of $Q \in \mathfrak{T}^s(\mathcal{G}, \mathcal{P}_y, \mathcal{P}_u)$ that are not constrained to be zero. Then

$$Q(z) \in \mathfrak{T}^s(\mathcal{G}, \mathcal{P}_y, \mathcal{P}_u) \Longleftrightarrow \mathbf{vec}(Q(z)) = H(z) S(z)$$

for some $S(z) \in \mathcal{RH}_\infty^{a \times 1}$, where $H(z)$ contains the delay and sparsity constraints imposed by the set $\mathfrak{T}^s(\mathcal{G}, \mathcal{P}_y, \mathcal{P}_u)$. Using the results of vectorization, we get that

$$\begin{aligned} &\|T_{11}(z) + T_{12}(z) Q(z) T_{21}(z)\|_2 \\ &\quad = \|\mathbf{vec}\left(T_{11}(z) + T_{12}(z) Q(z) T_{21}(z)\right)\|_2 \\ &\quad = \left\|\mathbf{vec}\left(T_{11}(z)\right) + \left(T_{21}(z)^t \otimes T_{12}(z)\right) \mathbf{vec}\left(Q(z)\right)\right\|_2 \\ &\quad = \left\|\mathbf{vec}\left(T_{11}(z)\right) + \left(T_{21}(z)^t \otimes T_{12}(z)\right) H(z) S(z)\right\|_2. \end{aligned}$$

Consider the following example where all the subsystems are SISO for simplicity:

$$\Phi(z) = \begin{bmatrix} h_1 & h_2 \\ h_3 & h_4 \end{bmatrix} - \begin{bmatrix} u_1 & u_2 \\ u_3 & u_4 \end{bmatrix} \begin{bmatrix} q_1 & q_2 \\ q_3 & q_4 \end{bmatrix} \begin{bmatrix} v_1 & 0 \\ v_3 & v_4 \end{bmatrix}$$

with the constraints $q_2 = 0$ and $q_3 = z^{-1} s_3$, $q_1 = s_1$, $q_4 = s_4$. Then

$$\mathbf{vec}\left(\Phi(z)\right) = \begin{bmatrix} h_1 \\ h_3 \\ h_2 \\ h_4 \end{bmatrix} - \begin{bmatrix} u_1 v_1 & u_2 v_1 z^{-1} & u_2 v_3 \\ u_3 v_1 & u_4 v_1 z^{-1} & u_4 v_3 \\ 0 & 0 & u_2 v_4 \\ 0 & 0 & u_4 v_4 \end{bmatrix} \begin{bmatrix} s_1 \\ s_3 \\ s_4 \end{bmatrix}.$$

Thus, we can pose the problem (35) as an unconstrained $\mathcal{H}_2$ problem

$$\begin{aligned} \min \quad & \left\|\mathbf{vec}\left(T_{11}(z)\right) + \left(T_{21}(z)^t \otimes T_{12}(z)\right) H(z) S(z)\right\|_2 \\ \text{subject to} \quad & S(z) \in \mathcal{RH}_\infty^{a \times 1} \end{aligned} \tag{36}$$

which can be solved using standard techniques. Let $S^\star(z)$ denote the solution of the optimization problem (36). Then the corresponding optimal $Q^\star(z)$ is given by $Q^\star(z) = \mathbf{vec}^{-1}(H(z) S^\star(z))$. Since $Q^\star(z) \in \mathfrak{T}^s(\mathcal{G}, \mathcal{P}_y, \mathcal{P}_u)$, we can obtain a realization $\tilde{Q} = (\tilde{A}_Q, \tilde{B}_Q, \tilde{C}_Q, \tilde{D}_Q) \in \mathfrak{S}^s(\mathcal{G}, \mathcal{P}_y, \mathcal{P}_u)$ using Theorem 1 such that $Q^\star(z) = \mathbf{tf}(\tilde{Q})$ and $\tilde{A}_Q$ is Schur-stable. The corresponding controller is given by $K^\star = \mathrm{F}_l(J, \tilde{Q})$, where J is given by (25). From Theorem 2, we can see that $K^\star$ thus designed is the optimal stabilizing controller implementable over the network $\mathcal{G}$ for the given plant P.

A. Example

In this section, we present a simple example to explain the various concepts and algorithms discussed in this paper. The example is chosen to show the many features of our modeling set-up and approach. Among them: 1) the dynamics of the nodes are heterogeneous. 2) The network is generic although small. It is not strongly connected, thus both sparsity and delay constraints are present in the plant input-output structure. 3) The optimal controller is unstable and reasonably complex, so we can use the full power of our approach to obtain a networked realization of it. We consider a strictly causal interaction of 3 sub-systems over a directed communication network represented by a directed pseudograph $\mathcal{G}$ given in Fig. 1. Let the 3 sub-systems $\{P_i\}_{i\in\{1,2,3\}}$ be expressed in their state-space representation as given below

$$P_1:\begin{bmatrix} x_1^+ \\ \hline z_1^+ \\ \hline y_1 \\ \hline \eta_{21}\end{bmatrix} = \left[\begin{array}{c|c|c|c} 1.5 & 1 & 1 & -2 \\ \hline -1 & 0 & 0 & 2 \\ \hline -1 & 0 & 0 & 2 \\ \hline 1 & 0 & 0 & 0 \end{array}\right] \begin{bmatrix} x_1 \\ \hline w_1 \\ \hline u_1 \\ \hline \nu_{12}\end{bmatrix}$$

$$P_2:\begin{bmatrix} x_2^+ \\ \hline z_2 \\ \hline y_2 \\ \hline \eta_{12} \\ \hline \eta_{32}\end{bmatrix} = \left[\begin{array}{c|c|c|c} 4 & 1 & 1 & -1 \\ \hline 1 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 \end{array}\right] \begin{bmatrix} x_2 \\ \hline w_2 \\ \hline u_2 \\ \hline \nu_{21}\end{bmatrix}$$

$$P_3:\begin{bmatrix} x_3^+ \\ \hline z_3 \\ \hline y_3 \\ \hline \eta_{03}\end{bmatrix} = \left[\begin{array}{c|c|c|c} 1.5 & 1 & 1 & 1 \\ \hline -1 & 0 & 0 & -1 \\ \hline -1 & 0 & 0 & -1 \\ \hline 0 & 0 & 0 & 0 \end{array}\right] \begin{bmatrix} x_3 \\ \hline w_3 \\ \hline u_3 \\ \hline \nu_{32}\end{bmatrix}$$

The network interconnection matrix, is as follows:

$$N:\begin{bmatrix} \nu_{12} \\ \hline \nu_{21} \\ \hline \nu_{32}\end{bmatrix} = \left[\begin{array}{c|cc|c} 0 & 1 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \end{array}\right] \begin{bmatrix} \eta_{21} \\ \hline \eta_{12} \\ \hline \eta_{32} \\ \hline \eta_{03}\end{bmatrix}$$

which is structured according to $\tilde{\mathcal{A}}(\mathcal{G}) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$.

By interconnecting the three sub-systems over the network, we get the interconnected system P with following state-space matrices:

$$\begin{bmatrix} x^+ \\ \hline z \\ \hline y \end{bmatrix} = \left[\begin{array}{ccc|ccc|ccc} 1.5 & -2 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 4 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1.5 & 0 & 0 & 1 & 0 & 0 & 1 \\ \hline -1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline -1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \begin{bmatrix} x \\ \hline w \\ \hline u \end{bmatrix}.$$

Note that A has eigenvalues outside the unit disc. For comparison purpose, the optimal centralized controller is just a gain

$$K_c = \begin{bmatrix} 1.1667 & -0.3333 & 0 \\ -1.6667 & -0.6667 & 0 \\ 0.1667 & 0.1667 & 1.5000 \end{bmatrix}.$$

The optimal centralized cost equals 3. Following Lemma 7 and Lemma 8, we obtain following matrices F and L so that $A+B_uF$ and $A+LC_y$ are Schur stable:

$$F = \begin{bmatrix} -1.0351 & 2.0702 & 0 \\ 1.9185 & -3.8371 & 0 \\ 0 & -1.1356 & -1.1356 \end{bmatrix}$$

$L = \mathrm{Diag}\,[\,1.0140 \quad -4.1139 \quad 1.0027\,]$.

Note that F is structured according to $\mathcal{A}(\mathcal{G})$ while L is block-diagonal. We can construct the following observer-based network realizable controller:

$$K_{\mathrm{nom}} = \left[\begin{array}{c|c} A+B_uF+LC_y & -L \\ \hline F & 0 \end{array}\right] \tag{37}$$

using the matrices F and L. Note that K_{nom} is a stabilizing controller implementable over $\mathcal{G}$. In this example, this sub-optimal distributed controller is unstable and gives a performance cost $|\mathrm{F}_l(P, K_{\mathrm{nom}})|_2 = 63.803$.

Then, following the formulation given in Section VII, we obtain the optimal distributed controller that is implementable over the given network. The performance cost $|T_{zw}|_2$ for this optimal controller is 8.8822. The order of the optimal distributed controller is 13 where the sub-systems K_1, K_2, and K_3 have order 4, 5 and 4, respectively. K_1 maps $[x_1^{K'}, y_1', \nu_{12}^{K'}]' \rightarrow [x_1^{K'}, u_1', \eta_{21}^{K'}]'$, K_2 maps $[x_2^{K'}, y_2', \nu_{21}^{K'}]' \rightarrow [x_2^{K'}, u_2', \eta_{12}^{K'}, \eta_{32}^{K'}]'$, and K_3 maps $[x_3^{K'}, y_3', \nu_{23}^{K'}]' \rightarrow [x_3^{K'}, u_3', \eta_{03}^{K'}]'$, see equation at the bottom of the next page.

The distributed controller uses controller order to compensate for the lack of full communication. However, the order of the resulting optimal controller can be quite large in general. Thus, the optimal cost provided by our optimal distributed controller can be used as a bound in designing sub-optimal reduced-order controllers realizable over the network.

Second, the optimal distributed controller is non-minimal but is stabilizable and detectable such that the closed-loop system is internally stable.

Last but not least, we note that the optimal distributed controller is unstable with two unstable poles at $-1.1157 \pm 1.3488i$. Realizing controllers like these, over a network, based only on their transfer functions is a problem in general.

In essence, this paper provides an optimal stabilizing network implementable controller and also provides a methodology to implement it over the given network even when the controller is unstable.

VIII. Conclusion

In this paper, we first gave a characterization of interconnected systems which are causal interactions of sub-systems over a delay-free and noiseless network in terms of the constraints followed by their state-space and input-output descriptions. Then, we introduced the notion of implementability and realizability of systems over delay-free and noiseless networks. We discussed the importance of the property of network implementability and made it a design requirement in the synthesis of distributed controllers. We provided constructive proofs to obtain network realizations for stable networked systems represented by structured transfer function matrices. We looked at the properties of interconnected systems that are network implementable and used these properties to parametrize the set of all-stabilizing network implementable controllers, under additional conditions. We then proposed a solution to the $\mathcal{H}_2$ optimal networked control problem for networked plants on the same network, and provided the optimal distributed controller implementable over the given network.

The results of this paper point to several directions for future research. In particular, more efficient network implementation and realization methods, the characterization of minimal networked realizations and the networked model-reduction for obtaining distributed controllers with lower order while maintaining network implementability property. As the results of the paper are rooted in the rich field of modern and robust control, we hope that they will lead to several extensions and new developments for networked systems.

Appendix

A. Proof of Theorem 1

Proof: The main idea of the proof is to show that if a stable $Q(z)$ satisfies the delay and sparsity constraints corresponding to $\mathfrak{T}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$, then the transfer function matrix can be realized over the delay-free and noiseless network $\mathcal{G}$ without introducing any hidden unstable modes that are not present in the stable transfer function $Q(z)$. There are obviously many more ways to get a network realization of $Q(z)$, but we present the following procedure to keep the proof simple. Since $Q(z) \in \mathfrak{T}(\mathcal{G}, \mathcal{P}_u, \mathcal{P}_y)$, $Q(z)$ is partitioned according to $(\mathcal{P}_y, \mathcal{P}_u)$. Let $\{Q_{ij}(z)\}_{i,j}$ be the partitions which are essentially the transfer function matrices mapping $U_j(z)$ to $Y_i(z)$. First, we separate these matrices into two categories following (6).

1) $Q_{ij}(z)$ is of the form $z^{-l(j,i)} H_{ij}(z)$ where $H_{ij}(z) \in \mathcal{R}_p$ and $l(j,i)$ is the length of a shortest path from node j to node i, i.e., the input of node j can affect the output of node i after $l(j,i)$ time steps. Recall that $l(i,i) \triangleq 0 \, \forall \, i \in \mathcal{V}$.
2) $Q_{ij}(z) = 0$, i.e., there exists no path from node j to node i on $\mathcal{G}$.

We do nothing about the zero transfer function matrices, while we further refine the classification in 1) based on the connectivity of the nodes.

In particular, we consider three cases: $l(j,i) = 0$, $l(j,i) = 1$, and $l(j,i) > 1$ For each case, we consider minimal realizations and define states corresponding to the mentioned nodes.

- When $l(j,i) = 0, \ j = i$, define states $x_{jj}(k)$ at node j

$$Q_{jj}(z): \quad \begin{array}{l} x_{jj}(k+1) = A_{jj} x_{jj}(k) + B_{jj} u_j(k) \\ y_{jj}(k) = C_{jj} x_j(k) + D_j u_j(k) \end{array}. \tag{38}$$

$$K_1 = \left[\begin{array}{cccc|c|c}
1.3153 & 0.3761 & 0.0162 & 0.0003 & 0.8504 & \\
-8.776 & -0.5699 & -0.1741 & -0.0029 & -8.776 & 0_{4,1}, \quad I_{4,4} \\
0.5454 & 2.0099 & 0.5923 & 0.0106 & 0.5454 & \\
0.0097 & -0.4995 & 0.9859 & -0.0224 & 0.0097 & \\
\hline
0.8292 & 0.3761 & 0.0162 & 0.0003 & 1.8643 & 1, \quad 0_{1,4} \\
\hline
3.2751 & 0.6224 & -0.0089 & -0.0003 & 0 & \\
-1.8388 & 0.6224 & -0.0089 & -0.0003 & 0 & \\
-8.7927 & 0 & 0 & 0 & 0 & 0_{6,1}, \quad 0_{6,4} \\
0.0658 & 0 & 0 & 0 & 0 & \\
0.0103 & 0 & 0 & 0 & 0 & \\
0 & 0.0741 & -0.0076 & 0.0009 & 0 &
\end{array}\right]$$

$$K_2 = \left[\begin{array}{ccccc|c|c}
-2.5944 & 1.0699 & -0.0249 & -0.0118 & 0 & 2.7573 & \\
-8.7927 & -0.6650 & -0.2102 & -0.1303 & 0 & 8.7927 & \\
0.0658 & 1.9547 & 0.0814 & 0.0802 & 0 & -0.0658 & 0_{5,1}, I_{5,5} \\
0.0103 & 0.5032 & 1.0182 & 0.5836 & 0 & -0.0103 & \\
0 & 0 & 0 & 0 & 0 & 0 & \\
\hline
-2.4806 & 1.0699 & -0.0249 & -0.0118 & 0 & -1.3565 & 1, 0_{1,5} \\
\hline
-1.6585 & 0.6675 & 0.0302 & 0.0207 & 0 & 0 & \\
-1.6306 & 0.6675 & 0.0302 & 0.0207 & 0 & 0 & \\
17.5520 & 0 & 0 & 0 & 0 & 0 & 0_{5,1}, 0_{5,5} \\
-0.0194 & 0 & 0 & 0 & 0 & 0 & \\
-1.091 & 0 & 0 & 0 & 0 & 0 & \\
\hline
0.2219 & -0.0909 & -0.0714 & 0.0129 & -1 & 0 & \\
0.2192 & -0.0909 & -0.0714 & 0.0129 & -1 & 0 & 0_{3,1}, 0_{3,5} \\
-8.6785 & 0 & 0 & 0 & 0 & 0 &
\end{array}\right]$$

$$K_3 = \left[\begin{array}{cccc|c|ccc}
0.7192 & 0.0689 & 0.0018 & -0.0366 & 0.3548 & 0 & 1 & 0 \\
-8.6785 & -0.6965 & 0.3506 & -0.0258 & -8.6785 & 0 & 0 & 1 \\
0 & -1.4659 & 0.7539 & -0.0579 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.2499 & -0.0575 & 0 & 0 & 0 & 0 \\
\hline
0.2219 & 0.0689 & 0.0018 & -0.0366 & 1.3575 & 1 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]$$

- When $l(j,i)=1, j \in \mathcal{N}_i^- \setminus \{i\}$ or $i \in \mathcal{N}_j^+ \setminus \{j\}$, define states $x_{ij}(k)$ at node j

$$Q_{ij}(z): \quad \begin{array}{l} x_{ij}(k+1) = A_{ij}x_{ij}(k) + B_{ij}u_j(k) \\ y_{ij}(k) = C_{ij}x_{ij}(k) \end{array}. \tag{39}$$

- When $l(j,i) \geq 2$, let a shortest path be given by $p(j,i) = m_{ij}^0 m_{ij}^1 \ldots m_{ij}^{l(j,i)}$, where $m_{ij}^0 = j$ and $m_{ij}^{l(j,i)} = i$, i.e., the path starts at node j and terminates at node i and m_{ij}^r for $r = 1, \ldots, l(j,i)-1$ are intermediate nodes. In this case, we define states at each node on the path as follows:

$$z^{-1}H_{ij}(z): \quad \begin{array}{l} x_{ij}^0(k+1) = A_{ij}x_{ij}^0(k) + B_{ij}u_j(k) \\ y_{ij}^0(k) = C_{ij}x_{ij}^0(k) \end{array}. \tag{40}$$

Note that states $x_{ij}^0(k)$ are defined at node j and the outputs $y_{ij}^0(k)$ are passed to node m_{ij}^1, i.e., the first node in the path from j to i. At nodes m_{ij}^r, $r = 1, \ldots, l(j,i)-1$, we define states $x_{ij}^r(k)$ corresponding to one-step delay systems

$$z^{-1}: \quad \begin{array}{l} x_{ij}^r(k+1) = y_{ij}^{r-1}(k) \\ y_{ij}^r(k) = x_{ij}^r(k). \end{array} \tag{41}$$

We denote the state vector corresponding to each node i to be $x_i(k)$, which is formed by appending the states $x_{ji}(k)\ \forall j \in \mathcal{N}_i^+$, $x_{ab}^r(k)$ whenever $m_{ab}^r = i$, i.e., when node i is an intermediate node of a shortest path from some node b to some other node a. Similarly, the network output vector $\eta_i(k)$ is formed by appending $y_{ji}(k)\ \forall j \in \mathcal{N}_i^+ \setminus \{i\}$ and $y_{ab}^r(k)$ (when $m_{ab}^r = i$), while the network input vector $\nu_i(k)$ is formed by appending $y_{ij}(k)\ \forall j \in \mathcal{N}_i^- \setminus \{i\}$ and $y_{ab}^{r-1}(k)$ (when $m_{ab}^r = i$). Note that the network inputs defined at node i do not instantaneously affect the network outputs at node i.

At node i, the output $y_i(k)$ is given by

$$y_i(k) = y_{ii}(k) + \sum_{j \in \mathcal{N}_i^- \setminus \{i\}} y_{ij}(k) + \sum_{j: l(j,i) \geq 2} y_{ij}^{l(j,i)-1}(k). \tag{42}$$

Thus, we can define n sub-systems, $\{\tilde{Q}_i\}_i$, each with local states $x_i(k)$, local inputs $u_i(k)$, local outputs $y_i(k)$, network inputs $\nu_i(k)$ and network outputs $\eta_i(k)$. Following the state-space equations (38)–(41), concerning these states, inputs and outputs at each node, we can see that $x_i(k+1)$ and $y_i(k)$ are linear functions of $x_i(k)$, $u_i(k)$ and $\nu_i(k)$ while $\eta_i(k)$ is only a function of $x_i(k)$. Thus, the n sub-systems $\{\tilde{Q}_i\}_i$ satisfy the structure given in (9) with $D_i^{\eta u} = 0$ and $D_i^{\eta\nu} = 0$ while the network inputs and network outputs satisfy (14). Thus $\{\tilde{Q}_i\}_i$ interacting over the zero-delay network $\mathcal{G}$ represents the network realization of the given $Q(z)$.

We show that the network realization thus obtained is also asymptotically stable and does not contain any internal unstable modes due to the introduction of additional states in the realization. To check the system stability, we consider the zero-input system by assuming $u_i(k) = 0\ \forall i, k$.

First, we shall separate the states defined earlier into two categories. The first category consists of the states corresponding to the transfer function matrices $Q_{ij}(z), \forall i \in \mathcal{V}, j \in \mathcal{N}_i^-$. This set of states can be written as $X_1(k) = \mathbf{vert}[x_{ij}(k)]_{i \in \mathcal{V}, j \in \mathcal{N}_i^-}$. From the state-space equations corresponding to these states, we get

$$X_1(k+1) = \mathbf{diag}[A_{ij}]_{i \in \mathcal{V}, j \in \mathcal{N}_i^-} X_1(k) \tag{43}$$

when $u_i(k) = 0$ for all i, k

The second category consists of the states corresponding to all the $Q_{ij}(z)$ when $l(j,i) \geq 2$. For example, assume that a shortest path from node j to node i has length greater than 1. Then

$$p(j,i) = m_{ij}^0 m_{ij}^1 \ldots m_{ij}^{l(j,i)}$$

where $l(j,i) \geq 2$, $m_{ij}^0 = j$ and $m_{ij}^{l(j,i)} = i$. Corresponding to this path, the states earlier defined are $x_{ij}^0(k), x_{ij}^1(k), \ldots, x_{ij}^{l(j,i)-1}(k)$. Let us define

$$X_{ij}(k) = \mathbf{vert}\left[x_{ij}^r(k)\right]_{r \in \{0, \ldots, l(j,i)-1\}}$$

corresponding to the path $p(j,i)$. From the state-space equations corresponding to these states, we can see that

$$X_{ij}(k+1) = \begin{bmatrix} A_{ij} & & & & \\ C_{ij} & 0 & & & \\ & I & 0 & & \\ & & I & 0 & \\ & & & & \ddots \end{bmatrix} X_{ij}(k). \tag{44}$$

Define $X_2(k) = \mathbf{vert}[X_{ij}(k)]_{\{i,j: 2 \leq l(j,i) < n\}}$ as the set of states corresponding to $Q_{ij}(z)$ when $l(j,i) \geq 2$. Note that $X_1(k)$ and $X_2(k)$ constitute all the states defined corresponding to the n sub-systems $\{\tilde{Q}_i\}_i$. From (43) and (44), we can see that the A- matrix corresponding to the dynamics of $\begin{bmatrix} X_1(k) \\ X_2(k) \end{bmatrix}$ is block lower triangular with $\{A_{ij}\}_{i,j}$ on the diagonal and the rest of the diagonal terms being zero.

By hypothesis, $Q(z)$ is BIBO stable which implies that $\{Q_{ij}(z)\}_{i,j}$ are all BIBO stable, which in turn implies that $\{H_{ij}(z)\}_{i,j}$ are all BIBO stable. Note that, we assumed minimal realizations of $H_{ij}(z)$ in (38)–(40) which implies that the matrices $\{A_{ij}\}_{i,j}$ are all Schur-stable. Thus, we can see that the A-matrix of the network realization also has its eigenvalues strictly inside the unit disc. This implies that the network realization corresponding to the n sub-systems $\{\tilde{Q}_i\}_i$ interacting over the network $\mathcal{G}$ is stabilizable and detectable even after introducing more states than required to minimally realize the transfer function matrix $Q(z)$. ∎

B. *Example for Realizing a Stable Transfer Function Matrix Over a Given Network*

In this example, we consider realizing a given transfer function as a strictly causal interaction of three sub-systems over a delay-free and noiseless network shown in Fig. 1(b). So, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{1,2,3\}$ and $\mathcal{E} = \{(1,1),(1,2),(2,1),(2,2),(2,3),(3,3)\}$. Let the transfer function of a stable interconnected system be given by

$$Q(z) = \begin{bmatrix} \frac{z+1}{z-0.5} & \frac{0.5}{z-0.8} & 0 \\ \frac{-0.1}{z-0.5} & \frac{z+0.1}{z-0.1} & 0 \\ \frac{1}{(z-0.1)(z-0.8)} & \frac{0.3}{z-0.8} & \frac{z-0.2}{z-0.5} \end{bmatrix}. \tag{45}$$

Note that (45) satisfies the delay and sparsity constraints corresponding to the causal network $\mathcal{G}$. Following the notation from Theorem 1, we write the minimal state-space realizations:

$$Q_{11}(z)=\frac{z+1}{z-0.5}\rightarrow\begin{bmatrix}x_{11}^{+}\\ \hline y_{11}\end{bmatrix}=\left[\begin{array}{c|c}0.5 & 1\\ \hline 1.5 & 1\end{array}\right]\begin{bmatrix}x_{11}\\ \hline u_1\end{bmatrix}$$
$$Q_{12}(z)=\frac{0.5}{z-0.8}\rightarrow\begin{bmatrix}x_{12}^{+}\\ \hline y_{12}\end{bmatrix}=\left[\begin{array}{c|c}0.8 & 1\\ \hline 0.5 & 0\end{array}\right]\begin{bmatrix}x_{12}\\ \hline u_2\end{bmatrix}$$
$$Q_{21}(z)=\frac{-0.1}{z-0.5}\rightarrow\begin{bmatrix}x_{21}^{+}\\ \hline y_{21}\end{bmatrix}=\left[\begin{array}{c|c}0.5 & 0.25\\ \hline -0.4 & 0\end{array}\right]\begin{bmatrix}x_{21}\\ \hline u_1\end{bmatrix}$$
$$Q_{22}(z)=\frac{z+0.1}{z-0.1}\rightarrow\begin{bmatrix}x_{22}^{+}\\ \hline y_{22}\end{bmatrix}=\left[\begin{array}{c|c}0.1 & 0.5\\ \hline 0.4 & 1\end{array}\right]\begin{bmatrix}x_{22}\\ \hline u_2\end{bmatrix}$$
$$z^{-1}H_{31}(z)=zQ_{31}(z)=\frac{z}{(z-0.1)(z-0.8)}$$
$$\rightarrow\begin{bmatrix}x_{31}^{0+}\\ \hline y_{31}^{0}\end{bmatrix}=\left[\begin{array}{cc|c}0.9 & -0.32 & 1\\ 0.25 & 0 & 0\\ \hline 1 & 0 & 0\end{array}\right]\begin{bmatrix}x_{31}^{0}\\ \hline u_1\end{bmatrix}$$
$$z^{-1}\rightarrow\begin{bmatrix}x_{31}^{1+}\\ \hline y_{31}^{1}\end{bmatrix}=\left[\begin{array}{c|c}0 & 1\\ \hline 1 & 0\end{array}\right]\begin{bmatrix}x_{31}^{1}\\ \hline y_{31}^{0}\end{bmatrix}$$
$$Q_{32}(z)=\frac{0.3}{z-0.8}\rightarrow\begin{bmatrix}x_{32}^{+}\\ \hline y_{32}\end{bmatrix}=\left[\begin{array}{c|c}0.8 & 0.5\\ \hline 0.6 & 0\end{array}\right]\begin{bmatrix}x_{32}\\ \hline u_2\end{bmatrix}$$
$$Q_{33}(z)=\frac{z-0.2}{z-0.5}\rightarrow\begin{bmatrix}x_{33}^{+}\\ \hline y_{33}\end{bmatrix}=\left[\begin{array}{c|c}0.5 & 0.5\\ \hline 0.6 & 1\end{array}\right]\begin{bmatrix}x_{33}\\ \hline u_3\end{bmatrix}.$$

In the graph $\mathcal{G}$, the shortest path (with length 2) from node 1 to node 3 is given by $1\rightarrow 2\rightarrow 3$ and the corresponding states are defined by $x_{31}^{0}(k)$ and $x_{31}^{1}(k)$. Following the proof of Theorem 1, we define state vectors corresponding to each node to be:

$$x_1=\begin{bmatrix}x_{11}\\ x_{21}\\ x_{31}^{0}\end{bmatrix},\quad x_2=\begin{bmatrix}x_{12}\\ x_{22}\\ x_{32}\\ x_{31}^{1}\end{bmatrix},\quad x_3=x_{33}.$$

The outgoing messages at each node are given by

$$\eta_1=\eta_{21}=\begin{bmatrix}y_{21}\\ y_{31}^{0}\end{bmatrix},\ \eta_2=\begin{bmatrix}\eta_{12}\\ \hline \eta_{32}\end{bmatrix}=\begin{bmatrix}y_{12}\\ \hline y_{32}\\ y_{31}^{1}\end{bmatrix},\ \eta_3=\eta_{03}$$

and the outputs at each node for each k are given by

$$y_1=y_{11}+y_{12}$$
$$y_2=y_{21}+y_{22}$$
$$y_3=y_{31}^{1}+y_{32}+y_{33}.$$

Since the network $\mathcal{G}$ is delay-free and noiseless, the incoming message vectors at each node are given by

$$\nu_1=\nu_{12}=\eta_{12},\quad \nu_2=\nu_{21}=\eta_{21},\quad \nu_3=\nu_{32}=\eta_{32}$$

which implies that the network interconnection matrix N is given by

$$\begin{bmatrix}\nu_1\\ \hline \nu_2\\ \hline \nu_3\end{bmatrix}=\left[\begin{array}{cc|ccc|c}0 & 0 & 1 & 0 & 0 & 0\\ \hline 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0\\ \hline 0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0\end{array}\right]\begin{bmatrix}\eta_1\\ \hline \eta_2\\ \hline \eta_3\end{bmatrix}. \tag{46}$$

Note that N is a static gain matrix which is structured according to $\begin{bmatrix}0 & 1 & 0\\ 1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}$ and partitioned according to $(\mathcal{P}_\nu,\mathcal{P}_\eta)$ where $\mathcal{P}_\nu=(1,2,2)$ and $\mathcal{P}_\eta=(2,3,1)$.

Using the state-space matrices of $Q_{ij}(z)$, the dynamics at each node i are defined as a sub-system $\tilde{Q}_i$ given by equation at the bottom of the page. Note that the procedure presented in the proof of Theorem 1 can lead to extra repeated stable eigenvalues. A minimal realization should be performed at each node, afterward. We leave the development of more efficient procedures to future research.

$$\tilde{Q}_1:\begin{bmatrix}x_{11}^{+}\\ x_{21}^{+}\\ x_{31}^{0+}\\ \hline y_1\\ \hline \eta_{21}\end{bmatrix}=\left[\begin{array}{cccc|c|c}0.5 & 0 & 0 & 0 & 1 & 0\\ 0 & 0.5 & 0 & 0 & 0.25 & 0\\ 0 & 0 & 0.9 & -0.32 & 1 & 0\\ 0 & 0 & 0.25 & 0 & 0 & 0\\ \hline 1.5 & 0 & 0 & 0 & 1 & 1\\ \hline 0 & -0.4 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0\end{array}\right]\begin{bmatrix}x_{11}\\ x_{21}\\ x_{31}^{0}\\ \hline u_1\\ \hline \nu_{12}\end{bmatrix}$$

$$\tilde{Q}_2:\begin{bmatrix}x_{12}^{+}\\ x_{22}^{+}\\ x_{32}^{+}\\ x_{31}^{1+}\\ \hline y_2\\ \hline \eta_{12}\\ \hline \eta_{32}\end{bmatrix}=\left[\begin{array}{cccc|c|cc}0.8 & 0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0.1 & 0 & 0 & 0.5 & 0 & 0\\ 0 & 0 & 0.8 & 0 & 0.5 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 1\\ \hline 0 & 0.4 & 0 & 0 & 1 & 1 & 0\\ \hline 0.5 & 0 & 0 & 0 & 0 & 0 & 0\\ \hline 0 & 0 & 0.6 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0 & 0\end{array}\right]\begin{bmatrix}x_{12}\\ x_{22}\\ x_{32}\\ x_{31}^{1}\\ \hline u_2\\ \hline \nu_{21}\end{bmatrix}$$

$$\tilde{Q}_3:\begin{bmatrix}x_{33}^{+}\\ \hline y_3\\ \hline \eta_{03}\end{bmatrix}=\left[\begin{array}{c|c|cc}0.5 & 0.5 & 0 & 0\\ \hline 0.6 & 1 & 1 & 1\\ \hline 0 & 0 & 0 & 0\end{array}\right]\begin{bmatrix}x_{33}\\ \hline u_3\\ \hline \nu_{32}\end{bmatrix}$$

Acknowledgment

The authors would like to thank the Associate Editor and the anonymous reviewers for their constructive and crucial comments in revising the paper.

References

[1] A. S. M. Vamsi and N. Elia, "Optimal realizable networked controllers for networked systems," in *Proc. Amer. Control Conf.*, Jun. 2011, pp. 336–341.

[2] N. Elia, J. Wang, and A. S. M. Vamsi, "Computing over unreliable communication networks," in *Information and Control in Networks. Lecture Notes in Control and Information Sciences*, vol. 450, G Como, B. Bernhardsson, and A. Rantzer, Eds. New York: Springer, 2014, pp. 241–275.

[3] B. Bamieh, F. Paganini, and M. A. Dahleh, "Distributed control of spatially-invariant systems," *IEEE Trans. Autom. Control*, vol. 47, no. 7, pp. 1091–1107, 2002.

[4] P. Voulgaris, G. Gianchini, and B. Bamieh, "Optimal decentralized controllers for spatially invariant systems," in *Proc. 39th IEEE Conf. Decision Control*, Dec. 2000, pp. 3763–3768.

[5] B. Bamieh and P. G. Voulgaris, "A convex characterization of distributed control problems in spatially invariant systems with communication constraints," *Syst. Control Lett.*, vol. 54, pp. 575–583, 2005.

[6] R. D'Andrea and G. E. Dullerud, "Distributed control design for spatially interconnected systems," *IEEE Trans. Autom. Control*, vol. 48, no. 9, pp. 1478–1495, Sep. 2003.

[7] M. Fardad and M. R. Jovanovic, "Design of optimal controllers for spatially invariant systems with finite communication speed," *Automatica*, vol. 47, pp. 880–889, 2011.

[8] X. Qi, M. V. Salapaka, P. G. Voulgaris, and M. Khammash, "Structured optimal and robust control with multiple criteia: A convex solution," *IEEE Trans. Autom. Control*, vol. 49, no. 10, pp. 1623–1640, Oct. 2004.

[9] P. G. Voulgaris, "A convex characterization of classes of problems in control with specific interaction and communication structures," in *Proc. Amer. Control Conf.*, Jun. 2001, pp. 3128–3133.

[10] M. Hovd and S. Skogestad, "Control of symmetrically interconnected systems," *Automatica*, vol. 30, no. 6, pp. 957–973, 1994.

[11] P. Shah, "A partial order approach to decentralized control," Ph.D. dissertation, MIT, Cambridge, MA, Jun. 2011.

[12] M. Rotkowitz and S. Lall, "A characterization of convex problems in decentralized control," *IEEE Trans. Autom. Control*, vol. 51, no. 2, pp. 1984–1996, Feb. 2006.

[13] L. Lessard and S. Lall, "Reduction of decentralized control problems to tractable representations," in *Proc. IEEE Conf. Decision Control*, Dec. 2009, pp. 1621–1626.

[14] S. Sabau and N. Martins, "Youla-like parametrizations subject to qi subspace constraints," *IEEE Trans. Autom. Control*, vol. 59, no. 6, pp. 1411–1422, 2014.

[15] A. Gattami, "Optimal decisions with limited information," Ph.D. dissertation, Lund Univ., Lund, Sweden, 2007.

[16] P. Massioni and M. Verhaegen, "Distributed control for identical dynamically coupled systems: A decomposition approach," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 124–135, Jan. 2009.

[17] C. Langbort, R. S. Chandra, and R. D'Andrea, "Distributed control design for systems interconnected over an arbitrary graph," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1502–1519, Sep. 2004.

[18] A. S. M. Vamsi and N. Elia, "A sub-optimal approach to design distributed controllers realizable over arbitrary networks," in *Proc. 2nd IFAC Workshop Distrib. Estim. Control Netw. Syst.*, Sep. 2010. [Online]. Available: http://necsys2010.inrialpes.fr/files/2010/09/0082.pdf

[19] L. Lessard and S. Lall, "A state-space solution to the two-player decentralized optimal control problem," in *Proc. 49th Annu. Allerton Conf. Commun., Control, Comp.*, Allerton, IL, 2011, pp. 1559–1564.

[20] A. Lamperski and J. Doyle, "Dynamic programming solutions for decentralized state-feedback lqg problems with communication delays," in *Proc. Amer. Control Conf.*, 2012, pp. 6322–6327.

[21] E. D. Sontag, *Mathematical Control Theory. Deterministic Finite Dimensional Systems*. New York: Springer-Verlag, 1990.

[22] L. Lessard, M. Kristalny, and A. Rantzer, "On structured realizability and stabilizability of linear systems," in *Proc. Amer. Control Conf.*, 2013, pp. 5784–5790.

[23] A. S. M. Vamsi and N. Elia, "Design of distributed controllers realizable over arbitrary directed networks," in *Proc. IEEE 49th Conf. Decision Control*, 2010, pp. 4795–4800.

[24] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[25] M. C. De Oliveira, J. C. Geromel, and J. Bernussou, "Extended H_2 and H_∞ norm characterizations and controller parametrizations for discrete-time systems," *Int. J. Control*, vol. 75, no. 9, pp. 666–679, Jun. 2002.

[26] A. I. Zecević and D. D. Siljak, "Control design with arbitrary information structure constraints," *Automatica*, vol. 44, pp. 2642–2647, 2008.

[27] N. Elia and M. A. Dahleh, "A quadratic programming approach for solving the l1 multi-block problem," *IEEE Trans. Autom. Control*, vol. 43, no. 9, pp. 1242–1256, Sep. 1998.

Andalam Satya Mohan Vamsi was born in Kanchikacherla, India, in 1982. He received the B.Tech. and M.S. degrees in electrical engineering from the Indian Institute of Technology (IIT) Madras, India, in 2003 and 2007, respectively, and the Ph.D. degree in electrical engineering from Iowa State University (ISU), Ames, in 2012.

Nicola Elia (SM'14) received the Laurea degree in electrical engineering from the Politecnico of Turin, Turin, Italy, in 1987, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 1996.

He was a Control Engineer with Fiat Research Center, Torino, Italy, from 1987 to 1990. He was a Postdoctoral Associate at the Laboratory for Information and Decision Systems, MIT, from 1996 to 1999. Presently, he is Professor of Electrical and Computer Engineering at Iowa State University, Ames. His research interests include computational methods for controller design, communication systems with access to feedback, control with communication constraints, and networked systems.

Dr. Elia received the NSF CAREER Award in 2001.