Topological Conditions for In-Network Stabilization of Dynamical Systems

Miroslav Pajic, *Member, IEEE,* Rahul Mangharam, *Member, IEEE,* George J. Pappas, *Fellow, IEEE* and Shreyas Sundaram, *Member, IEEE*

***Abstract*—We study the problem of stabilizing a linear system over a wireless network using a simple *in-network* computation method. Specifically, we study an architecture called the "Wireless Control Network" (WCN), where each wireless node maintains a state, and periodically updates it as a linear combination of neighboring plant outputs and node states. This architecture has previously been shown to have low computational overhead and beneficial scheduling and compositionality properties. In this paper we characterize fundamental topological conditions to allow stabilization using such a scheme. To achieve this, we exploit the fact that the WCN scheme causes the network to act as a linear dynamical system, and analyze the coupling between the plant's dynamics and the dynamics of the network. We show that stabilizing control inputs can be computed in-network if the vertex connectivity of the network is larger than the geometric multiplicity of any unstable eigenvalue of the plant. This condition is analogous to the typical min-cut condition required in classical information dissemination problems. Furthermore, we specify equivalent topological conditions for stabilization over a wired (or point-to-point) network that employs network coding in a traditional way – as a communication mechanism between the plant's sensors and decentralized controllers at the actuators.**

***Index Terms*—Networked control systems, decentralized control, wireless sensor networks, structured systems, in-network control, network coding, cooperative control**

I. Introduction

With recent revolutions in sensor and actuator technologies, availability of powerful but inexpensive embedded computing and introduction of new multi-hop wireless network standards for industrial automation, control over wireless networks is becoming a disruptive technology. Traditional wired interconnections between the plant sensors, controllers and actuators can be replaced by wireless multi-hop mesh networks, yielding cost and space savings for the plant operator. These improvements have also enabled more efficient and robust means of communication, and the opportunity to move the computation of the control law within the network.

Despite this tremendous promise, the introduction of wireless communications into the feedback loop presents several challenges for real-time feedback control. For instance, delays may be introduced if a multi-hop wireless network is used to route information between the plant sensors, actuators and controllers. Furthermore, transmissions in the network must be scheduled carefully to avoid packet dropouts due to collisions between neighboring nodes. These issues can be detrimental to the goal of maintaining stability of the closed loop system if not explicitly accounted for, and substantial research has been devoted to understanding the performance limitations in such settings (e.g., [2], [3], [4]). These works typically adopt the convention of having one or more dedicated controllers or state estimators located in the system, and study the stability of the closed loop system assuming that the sensor-estimator and/or controller-actuator communication channels are unreliable (dropping packets with a certain probability, for example). For this standard architecture, shown in Fig. 1(a), the use of dedicated controllers imposes a routing requirement along one or more fixed paths through the network, along with strict end-to-end delay constraints to ensure stability [5].

Routing couples the communication, computation and control problems [6]. This introduces additional problems when the network is shared among control loops (i.e., a node may be involved in the feedback path for many plants), and new control loops are added at run-time. With standard architectures for control over wireless networks, it may be necessary to completely recompute the control algorithms, communication schedules, and computation schedules every time a new loop is added to the system. To avoid this complexity, it is necessary to derive a *composable* control scheme, where control loops can be easily added and a simple compositional analysis can be performed at run-time to ensure that a new loop does not affect the functioning of existing control loops. In order to do so, one requires an alternative to the routing-based approaches currently employed for control over wireless networks.

A. The Wireless Control Network

Motivated by the above issues, in a recent paper [7] we asked the following question: is it possible to do away with the standard "sensor → channel → controller/estimator → channel → actuator" architecture (Fig. 1(a)) and have the computation of the control law be performed *in-network*? In other words, is it possible to formulate a distributed algorithm for the (resource constrained) wireless nodes to follow so that the network *itself* acts as a controller for the plant?

To answer this question, we considered a setup where a network of wireless nodes is deployed in the proximity of a plant, with some nodes having access to the sensor

Manuscript received April 9, 2012; revised January 05, 2013. This work has been partially supported by the NSF-CNS 0931239 and NSF-MRI 0923518 grants. It has also been funded by a grant from the Natural Sciences and Engineering Research Council of Canada (NSERC). Some of the results in this paper were presented in preliminary form in [1].

M. Pajic, G. J. Pappas and R. Mangharam are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA 19014 (e-mail: {pajic, pappasg, rahulm}@seas.upenn.edu).

S. Sundaram is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, N2L 3G1 (e-mail: ssundara@uwaterloo.ca).

Digital Object Identifier 10.1109/JSAC.2013.130415.

0733-8716/13/$31.00 © 2013 IEEE

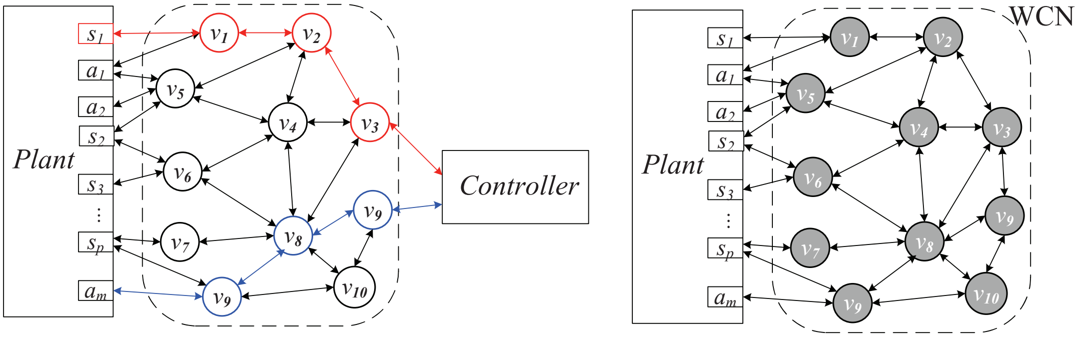


Fig. 1. (a) Standard architectures used for control over wireless network; Red links/nodes - routing data from the plant's sensors to the controller; Blue links/nodes - routing data from the controller to the actuators; (b) A multi-hop Wireless Control Network, where the network acts as a distributed controller.

measurements (outputs) of the plant, and some nodes placed within the listening range of the plant's actuators (as shown in Fig. 1(b)). To model resource constrained nodes, we assumed that each node is capable of maintaining only a limited internal state. We then presented a distributed algorithm in the form of a *linear iterative strategy* for each node to follow, where each node periodically updates its state to be a linear combination of the states of the nodes in its immediate neighborhood. The actuators of the plant also apply linear combinations of the states of the nodes in their neighborhood. Given a linear plant model and the network's topology, we devised a design-time procedure to derive the coefficients of the linear combinations for each node and actuator to apply in order to stabilize the plant. We showed that our method could also handle a sufficiently low rate of packet dropouts in the network to maintain mean square stability. We referred to this paradigm, where the computation of the control law is done *in-network* (i.e., in a distributed fashion by the wireless nodes), as a *Wireless Control Network* (WCN). The scheme has several benefits, including easy scheduling of wireless transmissions, compositional design, and the ability to handle geographically separated sensors and actuators. We illustrated the use of the WCN in industrial process control applications in [8].

While our previous work has established the feasibility of in-network computation for control, and provided numerical algorithms to obtain appropriate control laws, an important question remains unanswered: What fundamental topological conditions should the network satisfy to be able to stabilize a given plant? This question is the focus of this paper.

B. Topological Conditions For Stabilization Versus Information Transmission

The simple linear updates performed by each node in the WCN resembles the linear iterative algorithms used for distributed function calculation and consensus (e.g., [9], [10], [11]) and network coding (e.g., [12], [13], [14]). The key difference pertains to the *objective* of the network. Specifically, the goal of the WCN is not to get all nodes in the network to agree on a certain value, or to allow sink nodes to recover values injected into the network by source nodes. Instead, the objective is to provide a simple distributed scheme (suitable for implementation on resource constrained nodes), such that the resulting network dynamics facilitate the *stabilization* of the attached physical system.

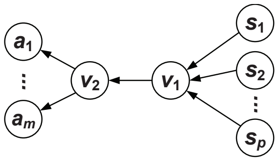


Fig. 2. A simple example of a wireless network between the plant's sensors $s_{1:p}$ and actuators $a_{1:m}$.

To illustrate the difference in the objectives, consider a plant with p sensors (measuring plant outputs), and m actuators (that apply control inputs to the plant to stabilize it), together with the network shown in Fig. 2. Node v_1 has access to the measurements provided by the p sensors at each time-step (or sampling period), and the actuators apply control inputs based upon information received from node v_2. Viewing the network in its traditional role as a transmission medium, the values from the p sensors (sources) would be expected to make their way to the actuators within one time-step. Each source injects one unit[1] of information per time-step into the network, and so the network needs a capacity of p units per time-step to deliver all of this information to the actuators. If the capacity of the edge (v_1, v_2) is only 1, the Min-Cut Max-Flow theorem [15] indicates that this objective is not achievable in this network, even without considering delay on any of the links.

However, the fact that this network is not capable of delivering all of the source information to all of the sinks at each time-step is not necessarily a cause for concern when the main objective is to *stabilize* the system. Specifically, the actuators do not necessarily need all of the source information, and the information received by the actuators at each time-step does not necessarily need to be a direct function of the information injected into the network at that time-step. Instead, the network only needs to supply the actuators with an appropriate set of inputs to apply at each time-step (perhaps after some additional computation at the actuators), and the fidelity of these inputs can be continually improved by the network based on the values received from the plant sensors. Given this (potentially relaxed) objective, what conditions should the network satisfy?

[1] In this paper, we consider the case of real-valued measurements, but in practice, the measurements and computations will be quantized to some finite precision.

C. Contributions of this Paper

We answer the questions posed above by characterizing network topologies that allow stabilization of a given linear dynamical system. We consider the WCN scheme, which causes the network to acts as a *linear dynamical system*, and study the coupling between the dynamics of the physical plant and the dynamics of the network. Our analysis draws upon ideas from *linear system theory*, *decentralized control theory* [16], [17], [18], [19], [20], [21], [22], and *structured system theory* [23], [24], which allows for the use of graph-theoretic tools to analyze dynamical systems. We show that for *stabilizable* and *detectable* plants, if the wireless network provides a sufficient number of vertex disjoint paths from certain plant sensors to certain plant actuators, then for the specific topology, there exists a WCN configuration (i.e., coefficients used in the linear iterative strategy) for which the closed-loop system is stable.

While this is reminiscent of the classical min-cut max-flow condition for information transmission, we prove that the size of the minimum network cut required to stabilize the network is not determined by the number of source nodes, as in typical information dissemination schemes, but rather by the *maximal geometric multiplicity of all unstable eigenvalues* of the plant. This reveals the interdependence between the dynamics of the physical process and the network topology. We also provide generic network conditions that are sufficient to stabilize *almost any* plant with a given structure – in the context of the example shown in Fig. 2, we show that a class of generic plants satisfying very loose structural conditions can be stabilized with this simple network.

Finally, we use ideas from the algebraic approach to network coding (e.g., [12], [25]) to specify equivalent topological conditions for the case of control over a wired (point-to-point) network, where network coding is used in its traditional role as a transmission mechanism between the plant's sensors and controllers located at the actuators.

D. Organization of the Paper

The rest of the paper is organized as follows. Section II provides our notation and a basic overview of linear systems. In Section III, we describe the WCN paradigm, along with its mathematical model. Section IV introduces concepts from decentralized control theory and structured system theory, which are used to derive topological conditions for stabilization of a generic class of linear systems with the WCN (Section V). In Section VI, we describe how to design a network with the minimal connectivity for stabilization. Section VII provides topological conditions for a numerically specified plant; this plant might fall within the measure zero set that is not covered by our analysis of generic systems. In Section VIII, we translate our results to the case when network coding over networks with point-to-point links is used to communicate information from sensors to controllers (placed at the actuators). Finally, we summarize our work in Section IX.

II. Notation and Terminology

We use $\mathbf{e}_i$ to denote the column vector (of appropriate size) with a 1 in its i-th position and 0's elsewhere. With $\mathbf{I}_N$ we denote the $N \times N$ identity matrix, while $\mathbf{I}$ denotes the identity matrix of appropriate dimensions. In addition, $\mathbf{A}'$ indicates the transpose of matrix $\mathbf{A}$. For a square matrix $\mathbf{Q}$, $\Lambda(\mathbf{Q})$ denotes the set of eigenvalues of $\mathbf{Q}$. The cardinality of a set $\mathcal{S}$ is denoted by $|\mathcal{S}|$, and for two sets $\mathcal{S}$ and $\mathcal{R}$, we use $\mathcal{S} \setminus \mathcal{R}$ to denote the set of elements in $\mathcal{S}$ that are not in $\mathcal{R}$. Finally, we denote the sets $\mathcal{M} = \{1, 2, ..., m\}$ and $\mathcal{P} = \{1, 2, ..., p\}$.

A. Linear Systems

Consider a system Σ of the form:

$$\begin{aligned} \mathbf{x}[k+1] &= \mathbf{A}\mathbf{x}[k] + \mathbf{B}\mathbf{u}[k] \\ \mathbf{y}[k] &= \mathbf{C}\mathbf{x}[k], \end{aligned} \tag{1}$$

where $\mathbf{x}[k] \in \mathbb{R}^n$ is the system state, $\mathbf{u}[k] \in \mathbb{R}^m$ is the input, and $\mathbf{y}[k] \in \mathbb{R}^p$ is the output, and the matrices are of appropriate dimensions. For convenience, we will denote the system as $\Sigma = (\mathbf{A}, \mathbf{B}, \mathbf{C})$.

The system is said to be *stable* if $\mathbf{x}[k] \to 0$ for any initial state $\mathbf{x}[0]$ when $\mathbf{u}[k] = 0$ for all k. The system is said to be *controllable* if for any initial state $\mathbf{x}[0]$ and for any final state $\mathbf{x}_f$, there exists an input sequence of finite length that transfers the state from $\mathbf{x}[0]$ to $\mathbf{x}_f$. The system is said to be *stabilizable* if for any initial state $\mathbf{x}[0]$, there is a sequence of inputs that causes $\mathbf{x}[k] \to 0$ as $k \to \infty$. The system is *observable* if for any unknown initial state $\mathbf{x}[0]$, there exists a finite integer $k_1 > 0$ such that the knowledge of the input and output sequences $\mathbf{u}[k]$ and $\mathbf{y}[k]$ from $k = 0$ to k_1 suffices to uniquely determine $\mathbf{x}[0]$. A generalization of observability is the concept of *detectability*, which says that $\mathbf{y}[0] = 0$ and $\mathbf{u}[k] = 0$ for all k implies that $\mathbf{x}[k] \to 0$ as $k \to \infty$.

B. Structured Linear Systems

A linear system of the form (1) is said to be *structured* if each entry in the system matrices is either a fixed zero or an independent free parameter [23]. A structured system Σ can be represented via a directed graph $\mathcal{G}_\Sigma = \{\mathcal{V}_\Sigma, \mathcal{E}_\Sigma\}$, which is sometimes referred to as a *structural graph*. The vertex set is given by $\mathcal{V}_\Sigma = \{\mathcal{X} \cup \mathcal{U} \cup \mathcal{Y}\}$ where $\mathcal{X} = \{x_1, ..., x_n\}$ denotes the set of state vertices, while $\mathcal{U} = \{u_1, ..., u_m\}$ and $\mathcal{Y} = \{y_1, ..., y_p\}$ denote the sets of input and output vertices, respectively. The edge set is given by $\mathcal{E}_\Sigma = \mathcal{E}_\mathbf{A} \cup \mathcal{E}_\mathbf{B} \cup \mathcal{E}_\mathbf{C}$ with $\mathcal{E}_\mathbf{A} = \{(x_i, x_j) | a_{ji} \neq 0\}$, $\mathcal{E}_\mathbf{B} = \{(u_i, x_j) | b_{ji} \neq 0\}$, $\mathcal{E}_\mathbf{C} = \{(x_i, y_j) | c_{ji} \neq 0\}$.

For a structured system, a simple path is called a *U-rooted path* if the path has its starting vertex in U. A number of mutually disjoint U-rooted paths is called a *U-rooted path family*. Similarly, a simple path that has its end vertex in Y is called a *Y-topped path*, while a number of mutually disjoint Y-topped paths is called a *Y-topped path family*.

We will be interested in properties of a structured system that can be inferred purely from the zero/nonzero structure of the system matrices. These properties will hold *almost everywhere* (i.e., the set of parameters for which the property does not hold has Lebesgue measure zero), and thus they are called *generic* properties [23]. Finally, two systems will be called structurally equivalent if they have the same number of states, inputs and outputs, and their system matrices have zeros in the same locations.

III. The Wireless Control Network

We consider the system presented in Fig. 1(b), where a wireless network is placed in the proximity of a system $\Sigma = (\mathbf{A},\mathbf{B},\mathbf{C})$ with state $\mathbf{x}\in\mathbb{R}^n$, input $\mathbf{u}\in\mathbb{R}^m$ and output $\mathbf{y}\in\mathbb{R}^p$. The output vector $\mathbf{y}[k]$ contains measurements of the plant state vector $\mathbf{x}[k]$ provided by the sensors from the set $\mathcal{S}=\{s_1,s_2,\ldots,s_p\}$, while the input vector $\mathbf{u}[k]$ corresponds to the signals applied to the plant by actuators from the set $\mathcal{A}=\{a_1,a_2,\ldots,a_m\}$.

The wireless network is described by a graph $\mathcal{G}=\{\mathcal{V},\mathcal{E}\}$, where $\mathcal{V}=\{v_1,v_2,\ldots,v_N\}$ is the set of N nodes and $\mathcal{E}\subseteq\mathcal{V}\times\mathcal{V}$ represents the radio connectivity (communication topology) in the network (i.e., edge $(v_j,v_i)\in\mathcal{E}$ if node v_i can receive information directly from node v_j). We define $\mathcal{V}_S\subseteq\mathcal{V}$ as the set of nodes that can receive information directly from at least one sensor, and $\mathcal{V}_A\subseteq\mathcal{V}$ as the set of nodes whose transmissions can be heard by at least one actuator. Furthermore, we define a new graph $\bar{\mathcal{G}}=\{\mathcal{V}\cup\mathcal{S}\cup\mathcal{A},\mathcal{E}\cup\mathcal{E}_{in}\cup\mathcal{E}_{out}\}$ that includes the initial graph $\mathcal{G}$, the plant's sensors and actuators and the edge sets:

$$\mathcal{E}_{out}=\left\{(s_l,v_i)\;\middle|\;\begin{array}{c}s_l\in\mathcal{S},v_i\in\mathcal{V}_S,\\ v_i\text{ can receive values from sensor } s_l\end{array}\right\},\tag{2}$$

$$\mathcal{E}_{in}=\left\{(v_i,a_l)\;\middle|\;\begin{array}{c}a_l\in\mathcal{A},v_i\in\mathcal{V}_A,\\ \text{actuator } a_l \text{ can receive values from } v_i\end{array}\right\}.\tag{3}$$

In [7], we proposed a method for distributed in-network computation of a stabilizing input sequence. The WCN scheme requires that each wireless node maintains a scalar[2] state and implements a simple, lightweight linear iterative procedure. At every time step (i.e., once every communication frame) each node in the network updates its state to be a linear combination of its previous state and the states of its neighbors. The update procedure of each node from the set $\mathcal{V}_S$ also includes a linear combination of the sensor measurements (i.e., plant outputs) from all sensors in its neighborhood. Denoting node v_i's state at time step k by $z_i[k]$, the update procedure is given by:[3]

$$z_i[k+1]=w_{ii}z_i[k]+\sum_{v_j\in\mathcal{N}_{v_i}}w_{ij}z_j[k]+\sum_{s_j\in\mathcal{N}_{v_i}}h_{ij}y_j[k].\tag{4}$$

Remark 1: For each node, the above update rule mimics the form of a traditional dynamical controller for system stabilization, with the difference that each node also views the states of adjacent nodes as inputs (and most nodes will not have access to the plant's outputs). Furthermore, the dimension of the state maintained by each node can be very small (e.g., a scalar), which is in contrast to the usual large state vectors maintained in typical controllers. As mentioned in the introduction, the WCN can also be viewed as a form of linear network coding [12], where each node repeatedly updates and transmits a value which is a linear combination of received values. Once again, the salient point is that the dynamics are introduced at each node to facilitate stabilization of the attached plant, and not to simply transmit information from one side of the network to the other. □

The original WCN scheme from [7] requires each plant input $u_i[k], i\in\{1,2,...,m\}$, to be a linear combination of values from the nodes in actuator a_i's neighborhood. In this work we generalize this and allow each actuator $a_i, (i\in\{1,2,...,m\})$ to maintain a (possibly) vector state[4] denoted by $\mathbf{z}_{a_i}[k]\in\mathbb{R}^{n_i}$ (for some $n_i\in\mathbb{Z}$). The procedure implemented by actuator a_i can be described as:

$$\begin{aligned}\mathbf{z}_{a_i}[k+1]&=\mathbf{W}_{a_i}\mathbf{z}_{a_i}[k]+\sum_{v_j\in\mathcal{N}_{a_i}}\mathbf{g}_{ij}z_j[k]\\ u_i[k]&=\mathbf{t}'_{a_i}\mathbf{z}_{a_i}[k]+\sum_{v_j\in\mathcal{N}_{a_i}}k_{ij}z_j[k],\end{aligned}\tag{5}$$

for some matrices $\mathbf{W}_{a_i}$, vectors $\mathbf{g}_{ij}$, $\mathbf{t}_{a_i}$ and scalars k_{ij}. Note that the above equation models the situation where the plant sensors and actuators are geographically separated, preventing the plant input from directly depending on any of the plant's outputs.

To specify the evolution of the states of all nodes and actuators in the network, we define at each time step k the node state vector $\mathbf{z}[k]=\begin{bmatrix}z_1[k]' & z_2[k]' & \ldots & z_N[k]'\end{bmatrix}'$ and the actuator state vector $\mathbf{z}_a[k]=\begin{bmatrix}\mathbf{z}_{a_1}[k]' & \mathbf{z}_{a_2}[k]' & \ldots & \mathbf{z}_{a_m}[k]'\end{bmatrix}'$. Therefore, these states evolve as:

$$\mathbf{z}[k+1]=\mathbf{W}\mathbf{z}[k]+\mathbf{H}\mathbf{y}[k]\ ,\tag{6}$$

$$\mathbf{z}_a[k+1]=\mathbf{W}_a\mathbf{z}_a[k]+\mathbf{G}\mathbf{z}[k].\tag{7}$$

In the above equations, the matrix $\mathbf{W}_a\in\mathbb{R}^{(\sum_{i=1}^m n_i)\times(\sum_{i=1}^m n_i)}$ is a block-diagonal matrix, while the matrices $\mathbf{W}\in\mathbb{R}^{N\times N}$, $\mathbf{H}\in\mathbb{R}^{N\times p}$ and $\mathbf{G}\in\mathbb{R}^{m\times N}$ have sparsity constraints imposed by the underlying WCN topology – the connections between the nodes in the network (for matrix $\mathbf{W}$), from the sensors to the nodes (for $\mathbf{H}$), and from the nodes to the actuators (for $\mathbf{G}$). Specifically, for all $i\in\{1,\ldots,N\}$, $w_{ij}=0$ if $v_j\notin\mathcal{N}_{v_i}\cup\{v_i\}$, $h_{ij}=0$ if $s_j\notin\mathcal{N}_{v_i}$, and $g_{ij}=0$ if $v_j\notin\mathcal{N}_{a_i}$.

Aggregating the node and actuator states into the network state vector $\hat{\mathbf{z}}=\begin{bmatrix}\mathbf{z}[k]' & \mathbf{z}_a[k]'\end{bmatrix}'$, the behavior of the network can be described as:

$$\begin{aligned}\hat{\mathbf{z}}[k+1]&=\underbrace{\begin{bmatrix}\mathbf{W} & \mathbf{0}\\ \mathbf{G} & \mathbf{W}_a\end{bmatrix}}_{\mathbf{W}_d}\hat{\mathbf{z}}[k]+\underbrace{\begin{bmatrix}\mathbf{H}\\ \mathbf{0}\end{bmatrix}}_{\mathbf{H}_d}\mathbf{y}[k]\\ \mathbf{u}[k]&=\mathbf{T}_a\mathbf{z}_a[k]+\mathbf{K}\mathbf{z}[k]=\underbrace{\begin{bmatrix}\mathbf{K} & \mathbf{T}_a\end{bmatrix}}_{\mathbf{G}_d}\hat{\mathbf{z}}[k],\end{aligned}\tag{8}$$

where $\mathbf{T}_a\in\mathbb{R}^{m\times m}$ is a block-diagonal matrix, and $\mathbf{K}\in\mathbb{R}^{m\times N}$ is a structured matrix with sparsity constraints imposed by the links from the network nodes to the actuators. From (8) we observe that the linear iterative strategy employed by all nodes and actuators causes the entire network to behave as a *structured linear system*. The dynamics of the system will be designed to stabilize the plant, and thus the wireless nodes and the actuators together act as a *dynamical compensator*.

[2] The small state size accounts for resource and computational constraints in the wireless nodes. The procedure can be extended to handle vector states at each node in a straightforward manner. However, the fundamental topological conditions for system stabilization, derived in this paper, will not change.

[3] The neighborhood $\mathcal{N}_v$ of a vertex v is with respect to the graph $\bar{\mathcal{G}}$.

[4] This scenario is motivated by practical reasons, since actuators are usually placed in fixed positions and are not power constrained, allowing them to utilize more powerful CPUs than the battery-operated wireless nodes.

Remark 2: Note that in the above scheme, the network operates at the same rate as the plant (i.e., the duration of the time-step k in (8) is the same as the duration of the time-step k for the plant Σ in (1)). In particular, there is *no routing* involved in this control scheme: information does not have to travel from the sensors to the actuators within one time-step. Instead, the dynamics of the network (encapsulated by its state vector and the update rule in (8)) allow the network to *generate* an appropriate stabilizing input $\mathbf{u}[k]$ at each time-step k. Meanwhile, the injected sensor measurements propagate through the network (via the nearest-neighbor rule specified in (4)) over time, updating the state and refining the control inputs that are generated. □

To describe the closed-loop system we denote with $\hat{\mathbf{x}}[k] = \begin{bmatrix}\mathbf{x}[k]' & \mathbf{z}[k]' & \mathbf{z}_a[k]'\end{bmatrix}'$ the overall system state that contains the state of the plant and states of the nodes and actuators. The overall closed-loop system evolves as:

$$\hat{\mathbf{x}}[k+1] = \begin{bmatrix} \mathbf{A} & \mathbf{B}\mathbf{G}_d \\ \mathbf{H}_d\mathbf{C} & \mathbf{W}_d \end{bmatrix} \begin{bmatrix}\mathbf{x}[k] \\ \hat{\mathbf{z}}[k]\end{bmatrix} \triangleq \hat{\mathbf{A}}\hat{\mathbf{x}}[k]. \qquad (9)$$

The closed-loop system described by (9) is stable if the matrix $\tilde{\mathbf{A}}_{\mathbf{d}} = \tilde{\mathbf{A}}_{\mathbf{d}}(\mathbf{W}_{\mathbf{d}}, \mathbf{H}_{\mathbf{d}}, \mathbf{G}_{\mathbf{d}})$ has all of its eigenvalues inside the unit circle. Since matrices $\mathbf{W}_{\mathbf{d}}, \mathbf{H}_{\mathbf{d}}, \mathbf{G}_{\mathbf{d}}$ are structured, choosing their values to obtain a stable $\hat{\mathbf{A}}$ can be cast in the form of a static output feedback problem with sparsity constraints on the gain matrix. This is a nonconvex problem (and hence difficult to solve in general), but various numerical procedures have been proposed in the literature (e.g., [26], [27]). In [7], we adapted some of these numerical procedures to find values for the nonzero WCN parameters so that the closed-loop system is stable, given a network topology and a predefined state size maintained by each node. In addition, a procedure similar to the ones from [7], [8] can be used to extract a stabilizing configuration[5] for the closed-loop system with unreliable communication links.[6] However, the proposed design-time procedure is iterative in nature, and convergence depends on the initialization point for the algorithm. Therefore, even in cases when a stabilizing configuration exists, the procedure might not be able to find it.

In this paper, we take a more fundamental approach and identify topological conditions on the network that *guarantee* the existence of a stabilizing configuration. To do this, we will use concepts from decentralized control theory pertaining to *fixed modes* of the linear system. Furthermore, since the WCN acts as a structured linear dynamical compensator, we use ideas from *structured systems theory* to obtain generic conditions that guarantee stabilization in this scenario.

IV. Decentralized Fixed Modes

In decentralized control systems, a set of non-interacting local controllers is used to control a dynamical system (plant); each of the controllers generates the appropriate plant inputs by observing *only a subset* of the plant's outputs. Due to these limitations imposed on each of the local controllers, it is possible that even a controllable and observable system can not be stabilized with the aforementioned setup. As shown in [16], the problem of decentralized control can be formulated as a *static output feedback control* problem, where the feedback matrix potentially has some sparsity constraints. Furthermore, [16] introduced the notion of fixed modes to derive conditions for the existence of a stabilizing set of decentralized controllers. The concept of fixed modes was generalized in [24] to handle arbitrary feedback patterns, and to enable a graph-theoretic analysis of the problem.

To formally define fixed modes, we consider a discrete-time system $\Sigma = (\mathbf{A}, \mathbf{B}, \mathbf{C})$ controlled by a set of m controllers where each controller is located at a different actuator, and has direct access to only a subset of the plant outputs.

Definition 1: The decentralized feedback patterns are specified as m sets $J_1, J_2, ..., J_m \subseteq \mathcal{P}$ ($\mathcal{P} = \{1, 2, ..., p\}$) such that for each $i \in \mathcal{M}$ ($\mathcal{M} = \{1, 2, ..., m\}$), $j \in J_i$ if and only if output y_j can be directly used to calculate input u_i. □

Using the above definition, m linear time-invariant dynamical feedback compensators are described as ($i = 1, ..., m$):

$$\begin{aligned} \mathbf{z}_i[k+1] &= \mathbf{F}_i\mathbf{z}_i[k] + \sum_{j\in J_i} \mathbf{q}_{ij} y_j[k] \\ u_i[k] &= \mathbf{h}_i'\mathbf{z}_i[k] + \sum_{j\in J_i} k_{ij} y_j[k], \end{aligned} \qquad (10)$$

where $\mathbf{z}_i \in \mathbb{R}^{n_i}$ is the controller's state vector, while matrix $\mathbf{F}_i$ and vectors $\mathbf{q}_i, \mathbf{h}_i$ are of appropriate dimensions. Based on the feedback patterns $J_1, J_2, \ldots, J_m$, we define the set

$$\mathbf{K}_f = \left\{\mathbf{K} \in \mathbb{R}^{m\times p} | k_{ij} = 0 \text{ if } j \notin J_i\right\}. \qquad (11)$$

Definition 2 ([16], [24]): For the system $\Sigma = (\mathbf{A}, \mathbf{B}, \mathbf{C})$, the set $\Lambda_f = \bigcap_{\mathbf{K}\in\mathbf{K}_f} \Lambda\left(\mathbf{A} + \mathbf{BKC}\right)$ is called the set of fixed modes with respect to the feedback structure constraints specified by $J_1, J_2, ..., J_m$. □

In words, the fixed modes are the eigenvalues of $\mathbf{A} + \mathbf{BKC}$ that remain fixed despite the choice of matrix $\mathbf{K} \in \mathbf{K}_f$. The following classical result explains the vital of fixed modes in the stabilizability analysis of linear dynamical systems.

Theorem 1 ([16]): The system Σ can be stabilized using the set of controllers defined in (10) if and only if all of its fixed modes are stable. □

Remark 3: The above result applies to the case where each of the decentralized controllers is a linear time-invariant (LTI) system. In general, it has been shown in the literature that one can obtain more relaxed conditions for decentralized stabilization by considering linear time-varying (LTV) controllers; these conditions are in terms of a concept known as *quotient fixed modes* [22], building on the notion of system *completeness* from [20], [21]. Furthermore, it has been shown that it is without loss of generality to consider LTV controllers for decentralized stabilization (i.e., if a given LTI system cannot be stabilized by a set of decentralized LTV controllers, then it cannot be stabilized by decentralized nonlinear controllers either) [22]. In this paper, we focus on LTI controllers of the form (5) at the actuators in order to develop a framework for stabilization over a WCN (with dynamics of the form (8)); the extension of our results to the general case of time-varying controllers is an avenue for future research. □

[5] In this work, matrices $\mathbf{W}_d$, $\mathbf{H}_d$ and $\mathbf{G}_d$ that satisfy the topological constraints and guarantee stability of $\hat{\mathbf{A}}$ are referred to as a *stabilizing configuration*.

[6] If the links can be modeled as independent Bernoulli processes, the stabilizing configuration guarantees mean square stability of the system.

For any subset $I \subseteq \mathcal{M}$ we define $J = \bigcup_{i \in \mathcal{M} \setminus I} J_i$. The following theorem characterizes the fixed modes of a given system with respect to the feedback pattern $J_1, J_2, \ldots, J_m$.

Theorem 2 ([18]): A complex number λ is a fixed mode of the system $\Sigma = (\mathbf{A}, \mathbf{B}, \mathbf{C})$ if and only if there exists a subset $I \subseteq \mathcal{M}$ such that

$$\operatorname{rank} \begin{bmatrix} \mathbf{A} - \lambda \mathbf{I} & \mathbf{B}_I \\ \mathbf{C}_J & \mathbf{0} \end{bmatrix} < n, \tag{12}$$

where $\mathbf{B}_I$ and $\mathbf{C}_J$ are the columns and rows of $\mathbf{B}$ and $\mathbf{C}$ indexed by the elements in sets I and J, respectively. □

Various other algebraic tests have been proposed to determine if a given system Σ has unstable fixed modes with respect to a given feedback pattern (e.g. [28], [17], [19]). These numerical tests are usually computationally intensive, and require calculation of the rank of a large number of matrices. In an effort to get away from numerical calculations and to analyze fixed modes of large-scale systems with uncertain parameters, a purely graph-theoretic test was provided in [24] to test whether a given system with a certain sparsity structure would have any fixed modes under a given feedback pattern.

As described in [17], there are two distinct reasons for a fixed mode. A fixed mode can either arise from a loss of rank due to a 'perfect cancellation' of the numerical parameters (which is a degenerate case), or it can be caused by deeper issues relating to the system structure. The latter set of fixed modes are called *structural fixed modes*.

Definition 3 ([17]): The system Σ has structural fixed modes with respect to the feedback pattern $J_1, J_2, \ldots, J_m$ if every system structurally equivalent to Σ has fixed modes with the same feedback pattern. □

As described in Section II-B, one can associate a graph $\mathcal{G}_\Sigma = \{\mathcal{V}_\Sigma, \mathcal{E}_\Sigma\}$ with the structure of a given system Σ. The graph can be augmented to capture a given feedback pattern $J_1, J_2 ..., J_m$ via a set of edges $\mathcal{E}_J = \{(y_j, u_i) | i \in \mathcal{M}, j \in J_i\}$. This produces the graph $\mathcal{G}_{\Sigma,J} = \{\mathcal{V}_\Sigma, \mathcal{E}_\Sigma \cup \mathcal{E}_J\}$. From this graphical representation of the closed-loop system, and using the approach from [24], we can state the following theorem that provides a graph-theoretic characterization of the conditions for nonexistence of structural fixed modes.

Theorem 3: The discrete-time system Σ with feedback pattern $J_1, J_2, \ldots, J_m$ has no structural fixed modes if and only if both of the following conditions hold:

i. Each state vertex $x_k \in \mathcal{X}$ is contained in a strong component of $\mathcal{G}_{\Sigma,J}$ that includes an edge from $\mathcal{E}_J$.
ii. There exists a set of disjoint cycles that covers all state vertices.

□

The second condition from the above theorem ensures that the system Σ does not have any fixed modes at zero. Although such modes are a concern for continuous-time systems, they are not an issue for stabilization of discrete-time plants (because fixed modes at zero are stable and would not violate Theorem 1). Hence, we state the following corollary.

Corollary 1: The discrete-time system Σ with feedback pattern $J_1, J_2, \ldots, J_m$ has no structural fixed modes (other than at the origin) if and only if each state vertex $x_k \in \mathcal{X}$ is contained in a strong component of $\mathcal{G}_{\Sigma,J}$ that includes an edge from $\mathcal{E}_J$. □

Since a system can have stable fixed modes outside of zero, the above corollary specifies sufficient (but not necessary) conditions for the existence of a set of stabilizing feedback controllers for almost every plant that has the given structure, with the given feedback pattern. A couple of caveats are in order. First, the theorem does not specify the size of the stabilizing controllers (i.e., the values for $n_i,\ i = 1, ..., m$, from (10)); only that sufficiently large controllers can be found at each actuator to jointly stabilize the system. This could be an issue when resource constrained processors are used as controllers (e.g., when wireless nodes in the WCN are used to compute the control laws). The second major caveat is that the existing analysis of decentralized feedback control systems assumes that each actuator has direct access to at least one of the plant outputs (i.e., the quantities $\mathbf{q}_{ij}$ and k_{ij} in (10) are nonzero). This leads to a nonempty set $\mathbf{K}_f$ in (11), and this assumption is utilized in the proof of sufficiency from [16] to show that all non-fixed modes can be stabilized.

These caveats prevent Corollary 1 from being directly used to analyze whether the system can be stabilized using a WCN. We would like the wireless nodes to maintain only small state vectors (ideally scalars). Even more importantly, from (5) it can be seen that plant inputs (actuators) do not have a direct connection from plant outputs. Instead, each node in the network uses the values received from other neighboring *nodes*, with only a few nodes incorporating sensor measurements in their updates. As a result, $\mathbf{K}_f$ from (11) contains only the zero matrix. Therefore, in this case, the role of fixed modes in stabilization over a network must be carefully studied. We do this in the subsequent sections.

V. Generic Topological Conditions for System Stabilization with Wireless Control Networks

In this section, we provide conditions for a given system to not have structural fixed modes when controlled using a WCN, where each node in the network maintains only a scalar state, and the actuator nodes maintain vector states.

We start our analysis by initially disregarding the effects of the actuators on the plant; i.e., we assume that at each time-step the plant actuators do not use transmissions from the nodes in the set $\mathcal{V}_A$ to actuate the plant (via (5)). This allows us to consider the plant $\Sigma = (\mathbf{A}, \mathbf{B}, \mathbf{C})$ and the WCN together as a linear system $\tilde{\Sigma}$, where the outputs of the plant are injected into the WCN (see Fig. 3). If we view the transmissions of the nodes in $\mathcal{V}_A$ as the output of the system $\tilde{\Sigma}$, the system can be specified as:

$$\hat{\mathbf{x}}[k+1] = \begin{bmatrix} \mathbf{x}[k+1] \\ \mathbf{z}[k+1] \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{HC} & \mathbf{W} \end{bmatrix}}_{\tilde{\mathbf{A}}} \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{z}[k] \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}}_{\tilde{\mathbf{B}}} \mathbf{u}[k],$$

$$\hat{\mathbf{y}}[k] = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{E}_{\mathcal{V}_A} \end{bmatrix}}_{\tilde{\mathbf{C}}} \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{z}[k] \end{bmatrix}. \tag{13}$$

Here, $\mathbf{E}_{\mathcal{V}_A} = \begin{bmatrix} \mathbf{e}_{i_1} & \mathbf{e}_{i_2} & \ldots & \mathbf{e}_{i_t} \end{bmatrix}'$ selects the state values from the set $\mathcal{V}_A = \{v_{i_1}, v_{i_2}, ..., v_{i_t}\}$ (where $t = |\mathcal{V}_A|$). In other words, the vector $\hat{\mathbf{y}}[k]$ contains the states transmitted by the wireless nodes closest to the actuators at time-step k.

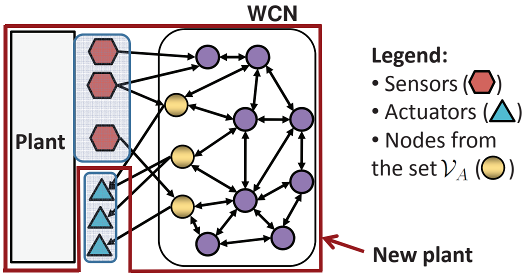


Fig. 3. Dynamical system $\tilde\Sigma$ that contains the dynamics of the plant and WCN; the states of the nodes from the set $\mathcal{V}_A$ represent the output of the system.

The structural graph $\mathcal{G}_{\tilde\Sigma} = (\mathcal{V}_{\tilde\Sigma}, \mathcal{E}_{\tilde\Sigma})$ of the system $\tilde\Sigma$ is obtained by composing the structural graph of the initial plant Σ and the network graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$:[7]

$$\mathcal{V}_{\tilde\Sigma} = \mathcal{X} \cup \mathcal{U} \cup \mathcal{V}, \quad \mathcal{E}_{\tilde\Sigma} = \mathcal{E}_{\mathbf{A}} \cup \mathcal{E}_{\mathbf{B}} \cup \mathcal{E} \cup \mathcal{E}_{\mathcal{O}}.$$

Recall that $\mathcal{X}$ is the set of state vertices (corresponding to the states of the plant), $\mathcal{U}$ is the set of p input vertices (corresponding to the actuators), and $\mathcal{V}$ is the set of vertices corresponding to the network nodes. The set $\mathcal{E}_{\mathbf{A}}$ represents the edges between state vertices (given by the matrix $\mathbf{A}$), and $\mathcal{E}_{\mathbf{B}}$ represents edges from the plant inputs to the states (given by the matrix $\mathbf{B}$). The set $\mathcal{E}$ represents the topology of the network, and the set $\mathcal{E}_{\mathcal{O}}$ captures how the state vertices influence the vertices in the wireless network. Specifically, the states of the plant affect the outputs of the plant (via the edge set $\mathcal{E}_{\mathbf{C}}$), and each plant output connects to one or more nodes (via the edge set $\mathcal{E}_{out}$ defined in (2)). As the output vertices simply pass the information about the state vertices through to the wireless network, we can remove the output vertices from the representation and introduce connections directly from the state vertices to the wireless vertices as follows:

$$\mathcal{E}_{\mathcal{O}} = \{(x_i, v_j) \in \mathcal{X} \times \mathcal{V}_S | \exists y_k \in \mathcal{Y}, (x_i, y_k) \in \mathcal{E}_{\mathbf{C}}, (y_k, v_j) \in \mathcal{E}_{out}\}.$$

Remark 4: Note that the edges from the set $\mathcal{E}_{\mathcal{O}}$ correspond to elements in the matrix $\mathbf{HC}$ from (13). To be able to reason about generic properties of a structured system, it is necessary for technical reasons to ensure that all of the system's parameters are *independent* [23]. Hence, we assume that the matrices $\mathbf{H}$ and $\mathbf{C}$ satisfy the property that either $\mathbf{H}$ has a single nonzero entry in each column (e.g., by having a dedicated node for each plant output), or $\mathbf{C}$ has a single nonzero entry in each row. This guarantees that each nonzero entry in $\mathbf{HC}$ will be an independent free parameter if each nonzero entry in $\mathbf{H}$ and $\mathbf{C}$ is an independent free parameter.

Furthermore, the matrix $\mathbf{E}_{\mathcal{V}_A}$ in (13) is a zero-one matrix with a single 1 in each row. While these are not independent free parameters, this does not affect the structural analysis because each row i can be effectively scaled by an independent free parameter p_i to produce the matrix $\mathbf{E}_{\mathcal{V}_A} = [p_1\mathbf{e}_{i_1} \;\; p_2\mathbf{e}_{i_2} \; ... \; p_t\mathbf{e}_{i_t}]'$; these parameters can then be taken into account while deriving the values for matrix $\mathbf{G}$ from (7). Thus, to simplify the notation and without loss of generality, we directly work with the system $\tilde\Sigma$ as specified in (13). □

The above representation of the system $\tilde\Sigma$ allows us to map the problem of stabilization using the WCN into a decentralized feedback control framework. Note that in (5), for each actuator a_i and each node $v_j \in \mathcal{N}_{a_i}$ there exists some row l of $\hat{y}[k]$ in (13) such that $z_j[k] = \hat{y}_l[k]$. Hence, the terms $\sum_{v_j \in \mathcal{N}_{a_i}} \mathbf{g}_{ij} z_j[k]$ and $\sum_{v_j \in \mathcal{N}_{a_i}} k_{ij} z_j[k]$ correspond to linear combinations of the system $\tilde\Sigma$'s outputs $\hat{\mathbf{y}}[k]$. In this setup, the overall system $\tilde\Sigma$ in (13) is to be controlled with a set of m decentralized feedback controllers described by (5). In addition, the feedback pattern is specified with the edge set $\mathcal{E}_{in}$ from (3) (i.e., in this case $\mathcal{E}_J = \mathcal{E}_{in}$). The key insight is the following: by having each wireless node run a linear strategy, the WCN and the plant together form a linear system $\tilde\Sigma$. Then, by viewing the transmissions of the wireless nodes closest to the actuators as the new 'outputs' of the system $\tilde\Sigma$, the problem of stabilizing the system with compensators at the actuators fits within the classical decentralized control formulation described in Section IV. Consequently, Corollary 1 can be applied to obtain the following topological condition that guarantees the existence of a stabilizing WCN configuration.

Theorem 4: Almost any system structurally equivalent to system $\Sigma = (\mathbf{A}, \mathbf{B}, \mathbf{C})$ can be stabilized with a WCN if for each plant state vertex $x_i \in \mathcal{X}$ in the structural graph $\mathcal{G}_{\tilde\Sigma_{in}} = (\mathcal{V}_{\tilde\Sigma}, \mathcal{E}_{\tilde\Sigma} \cup \mathcal{E}_{in})$ there exists a cycle that contains the state vertex $x_i \in \mathcal{X}$ and any WCN vertex from $\mathcal{V}$. □

Proof: Consider the graph $\mathcal{G}_{\tilde\Sigma} = (\mathcal{V}_{\tilde\Sigma}, \mathcal{E}_{\tilde\Sigma})$ of the structured system (13) composed of the plant and the WCN. For each plant state vertex $x_i \in \mathcal{X}$ in the structural graph $\mathcal{G}_{\Sigma}$, let A_i denote the set of input vertices from which x_i is reachable in the initial system, while $\mathcal{V}_{A_i}$ denotes the set of WCN nodes that are neighbors of the actuators in A_i. If for a plant state vertex x_i there exists a WCN state vertex $z_j \in \mathcal{V}_{A_i}$ reachable from x_i, then x_i belongs to a strong component with an edge from $\mathcal{E}_{in}$. Since this holds for all plant state vertices, if all network state vertices belong to a strong component that contains an edge from $\mathcal{E}_{in}$, Corollary 1 will be satisfied, and the system will not have structural fixed modes outside of the origin.

On the other hand, a fixed mode will be introduced with each WCN state vertex z_i that does not belong to a strong component in the graph $\mathcal{G}_{\tilde\Sigma_{in}} = (\mathcal{V}_{\tilde\Sigma}, \mathcal{E}_{\tilde\Sigma} \cup \mathcal{E}_{in})$ with an edge from $\mathcal{E}_{in}$ (this might happen if the network is disconnected). However, by setting to zero all the weights associated with the links outgoing from z_i, this WCN state vertex is effectively removed from the network. In this case, due to the state vertex z_i the system has a structured fixed mode in the origin. Thus, in both cases the closed-loop system does not have structured fixed-modes outside of zero, meaning that almost every system with this structure will be stabilizable using the WCN. ■

VI. Minimal Stabilizing Feedback Connections

In this section, we investigate the minimal connectivity that the WCN should provide to ensure that the conditions from the previous section hold.

For traditional decentralized continuous-time systems, [29] considered the problem of determining the minimal number of

[7] While $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ refers to the 'physical' graph, when all the nodes in the network maintain a scalar state there is a one-to-one correspondence between this graph and a structural graph of the WCN (viewed as a structured controller (6)). Therefore, we will also use $\mathcal{G}$ as a structural graph.

direct connections between plant outputs and inputs to ensure that the system does not have structured fixed modes. We will now present a simplified procedure for discrete-time control systems by leveraging the fact that fixed modes at zero do not cause problems for stabilization in discrete-time. Specifically, we determine a minimal set of feedback edges that guarantee the absence of nonzero structural fixed modes. We will then use this result in conjunction with our results from the previous section to infer properties that the WCN should satisfy in order to stabilize the plant.

Consider a system $\Sigma = (\mathbf{A}, \mathbf{B}, \mathbf{C})$. For all sets $I \subseteq \mathcal{M}$ and $J \subseteq \mathcal{P}$ we denote with $\mathbf{B}_I$ and $\mathbf{C}_J$ submatrices of $\mathbf{B}$ and $\mathbf{C}$ consisting of columns of $\mathbf{B}$ and rows of $\mathbf{C}$ with indices in I and J, respectively. A system $\Sigma_{IJ} = (\mathbf{A}, \mathbf{B}_I, \mathbf{C}_J)$ can be described with a graph $\mathcal{G}_{\Sigma_{IJ}} = \{\mathcal{V}_{\Sigma_{IJ}}, \mathcal{E}_{\Sigma_{IJ}}\}$, which can be obtained from $\mathcal{G}_\Sigma = \{\mathcal{V}_\Sigma, \mathcal{E}_\Sigma\}$ by keeping input vertices from the index set I and output vertices associated with set J. These sets are denoted by $\mathcal{U}_I$ and $\mathcal{Y}_J$, respectively. We will require the following results that specify a set of conditions for structural controllability/observability.

Theorem 5 ([23]): A structured system is structurally controllable (observable) if and only if each state vertex in the corresponding graph is the end (beginning) of a U-rooted (Y-topped) path, and there exists a disjoint union of a U-rooted (Y-topped) path family and a cycle family that covers all state vertices. □

The condition pertaining to disjoint paths and cycles in the above theorem is only to preclude uncontrollable and unobservable modes at zero. As with the case of fixed modes, these modes at the origin are not a major concern for discrete-time systems, and thus we present the following simplified tests for *structural stabilizability* and *detectability*.

Corollary 2: A structured system is structurally stabilizable (detectable) if each state vertex is the end (beginning) of a U-rooted (Y-topped) path. □

Definition 4: A stabilizable subset of the plant inputs (i.e., actuators) is a set $I \subseteq \mathcal{M}$ such that $(\mathbf{A}, \mathbf{B}_I)$ is structurally stabilizable. Similarly, a detectable subset of the outputs (i.e., sensors) is a set $J \subseteq \mathcal{P}$ for which $(\mathbf{A}, \mathbf{C}_J)$ is structurally detectable. □

For some stabilizable subsets I, it may be possible to find an even smaller stabilizable subset $I' \subset I$. Since we wish to investigate the minimal feedback connectivity requirements, we use the notion of essential input and output sets from [29].

Definition 5: A stabilizable subset I is called an essential input set if there is no structurally stabilizable (strict) subset $I' \subset I$. A detectable subset J is called an essential output set if there is no structurally detectable (strict) subset $J' \subset J$. □

Note that for a particular system $\Sigma = (\mathbf{A}, \mathbf{B}, \mathbf{C})$ there might exist several different essential input and output sets, with potentially different numbers of elements. We use essential input and output sets to determine the minimal number of feedback connections that would guarantee that a system does not have nonzero structural fixed modes. From Corollary 1, for essential input and output sets I and J, at least $\max(|I|, |J|)$ feedback connections have to be used. We now show that this number of feedback connections is also sufficient.

Theorem 6: For a structurally stabilizable and detectable system $\Sigma = (\mathbf{A}, \mathbf{B}, \mathbf{C})$, let I and J be an essential input and output set, respectively. Then the system can be stabilized by introducing $\max(|I|, |J|)$ feedback connections (directly between appropriate outputs and inputs). □

The proof (in Appendix A) defines Algorithm 1 that takes the sets I and J as input and creates $\max(|I|, |J|)$ feedback lines between output vertices from $\mathcal{Y}_J$ and input vertices from $\mathcal{U}_I$, which satisfy the conditions from Corollary 1.

We now apply these general results to the case where a WCN is used for control. As before, the key trick is to view the composition of the WCN and the plant as a new dynamical system. In this case, the set of nodes $\mathcal{V}_\mathcal{A}$ (in the neighborhood of the actuators) corresponds to the outputs of the new system. The new system will be structurally detectable if there exists a path between each essential plant output and a node from $\mathcal{V}_\mathcal{A}$. Therefore, we introduce the following results.

Definition 6: A detectable set of WCN nodes $\mathcal{V}_{\mathcal{DET}} \subseteq \mathcal{V}_\mathcal{A}$ is a set of nodes such that for each sensor s_j that corresponds to an output y_j from an essential output set J, there exists a path from s_j to a node from $\mathcal{V}_{\mathcal{DET}}$. □

Corollary 3: Consider a structurally stabilizable and detectable system $\Sigma(\mathbf{A}, \mathbf{B}, \mathbf{C})$ with essential input and output sets I and J. The system can be stabilized with a WCN described by a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ using $\max(|I|, |\mathcal{V}_{\mathcal{DET}}|)$ links between the nodes from a detectable set $\mathcal{V}_{\mathcal{DET}}$ and actuators corresponding to the essential input set I. □

The proof of the above corollary is readily obtained by noticing that if such a detectable set of nodes $\mathcal{V}_{\mathcal{DET}} \subseteq \mathcal{V}_\mathcal{A}$ exists, then due to structural detectability of the plant there would be a path from each plant state vertex to a vertex representing the state of a node from $\mathcal{V}_{\mathcal{DET}}$. Furthermore, all network nodes that do not have a path to at least one node from $\mathcal{V}_{\mathcal{DET}}$ can be disregarded as in the proof of Theorem 4 (by setting all related weights to zero). Hence, the 'new' system $\hat{\Sigma}$ that contains the plant and the network is structurally detectable. Similarly, it can be shown that the 'new' system is stabilizable and the proof follows from Theorem 6, by applying Algorithm 1.

However, there is a possibility that the feedback edges created by Algorithm 1 cannot be physically implemented, as it might cause an actuator to rely on a wireless node that is not actually in its neighborhood (e.g., if an actuator is outside of a node's communication range). The following corollary introduces a straightforward condition to preclude this case, and a requirement for designing WCNs that guarantee stabilization of almost all systems with a certain structure.

Corollary 4: Almost every structurally stabilizable and detectable system $\Sigma = (\mathbf{A}, \mathbf{B}, \mathbf{C})$ can be stabilized if the following conditions are met:

i. The WCN is strongly connected.
ii. There exists an essential output set with each sensor in the set connected to the network.
iii. There exists an essential input set where each actuator in the set is a neighbor of at least one network node.

□

The corollary follows from Theorem 4 since in a strongly connected network where each sensor (i.e., plant output) connects to at least one network node, there is a path from every sensor to every node, including all nodes from $\mathcal{V}_\mathcal{A}$.

VII. WCN Topology Design to Stabilize a Numerically Specified Plant

In the previous sections, we have been focused on designing a WCN for a plant from a purely structural perspective, without regard for the numerical values. This allowed us to characterize WCN properties that would guarantee stabilization of almost any plant having a certain structure. However, one may be interested in designing a WCN for a given (numerically specified) system $\Sigma = (\mathbf{A}, \mathbf{B}, \mathbf{C})$. If this system falls within the measure zero set that is not covered by the structural analysis, one has to be more careful in designing the WCN. Specifically, any plant that has nonzero eigenvalues of multiplicity larger than 1 will not be captured by the generic set [18], and we will show that the multiplicity of eigenvalues in the plant will require the WCN to contain *linkings* of a sufficiently large size.[8] To the best of our knowledge this is the first work that studies the interplay between numerically specified systems (with eigenvalues of multiplicity larger than one), and structured systems (where graph-theoretic analysis dominates). Previous approaches that used graph-theory to analyze numerical systems were limited to the cases where all eigenvalues have multiplicity equal to one (e.g., [19]).

Consider a WCN used to control a given (numerically specified) system $\Sigma = (\mathbf{A}, \mathbf{B}, \mathbf{C})$, where the pair $(\mathbf{A}, \mathbf{C})$ is detectable, and the pair $(\mathbf{A}, \mathbf{B})$ is stabilizable. Assuming for now that the plant actuators do not close the loop via the transmissions of nearby wireless nodes, the overall system $\tilde{\Sigma} = (\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}})$ (plant and wireless network) is given by (13). As in the previous sections, we consider the following problem. How should the WCN be designed to guarantee that a dynamic compensator can be designed at each actuator to stabilize the system, when each actuator only receives the transmissions of the wireless nodes in its neighborhood?

To answer this, for any actuator a_i, let $\mathcal{V}_{a_i}$ denote the WCN nodes whose transmissions can be heard by a_i. For any set $I \subseteq \mathcal{M}$, define $\mathcal{V}_{\mathcal{M}\setminus I} = \bigcup_{i \in \mathcal{M}\setminus I} \mathcal{V}_{a_i}$ as the set of all nodes that are in the neighborhood of actuators not in I. To show that the system (13) has no fixed modes with respect to the feedback structure $\mathcal{V}_{a_1}, ..., \mathcal{V}_{a_m}$, we use Theorem 2 to prove that for all unstable eigenvalues λ of the matrices $\mathbf{A}$ or $\mathbf{W}$, $\operatorname{rank}(\tilde{\mathbf{M}}_{I,F}(\lambda)) \geq n + N$ where

$$\tilde{\mathbf{M}}_{I,F}(\lambda) \triangleq \begin{bmatrix} \mathbf{A} - \lambda\mathbf{I} & \mathbf{0} & \mathbf{B}_I \\ \mathbf{HC} & \mathbf{W} - \lambda\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{E}_F & \mathbf{0} \end{bmatrix}. \qquad (14)$$

Here, $\mathbf{E}_F$ is a matrix with a single 1 in each row, selecting the portions of the WCN state vector $\mathbf{z}[k]$ corresponding to the nodes in $\mathcal{V}_{\mathcal{M}\setminus I}$. We start with the following lemma.

Lemma 1: For almost any choice of nonzero parameters in $\mathbf{W}$, a nonzero eigenvalue λ of $\mathbf{A}$ is a fixed mode of $\tilde{\Sigma} = (\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}})$ if and only if it is a fixed mode of the system $\bar{\Sigma} = (\mathbf{A}, \mathbf{B}, \mathbf{E}_{\mathcal{V}_A}(\mathbf{W} - \lambda\mathbf{I})^{-1}\mathbf{HC})$. □

Proof: For a structured square matrix $\mathbf{W}$ and for a finite set of nonzero complex numbers $\mathcal{L}$, the eigenvalues of $\mathbf{W}$ will all be different from the elements of $\mathcal{L}$ for almost any choice of parameters in $\mathbf{W}$ [18]. In particular, this implies that any nonzero eigenvalue λ of $\mathbf{A}$ will not be an eigenvalue of $\mathbf{W}$ (for almost any choice of free parameters). Then, for any $I \subseteq \mathcal{M}$, the matrix $\tilde{\mathbf{M}}_{I,F}$ from (14) has rank as shown in (15).

Therefore, λ is a fixed mode of $\tilde{\Sigma} = (\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}})$ with respect to $\mathcal{V}_{a_1}, ..., \mathcal{V}_{a_m}$ if and only if it is a fixed mode of $(\mathbf{A}, \mathbf{B}, \mathbf{E}_{\mathcal{V}_A}(\mathbf{W} - \lambda\mathbf{I})^{-1}\mathbf{HC})$, with respect to the feedback pattern $\mathcal{V}_{a_1}, ..., \mathcal{V}_{a_m}$. ■

Consider any set $I \subseteq \mathcal{M}$, and let $\operatorname{rank}\begin{bmatrix}\mathbf{A} - \lambda\mathbf{I} & \mathbf{B}_I\end{bmatrix} = n - d_I$, where d_I is a nonnegative integer. Thus, to ensure that λ is not a fixed mode of the system $\tilde{\Sigma}$, the matrix $\begin{bmatrix}\mathbf{E}_F(\mathbf{W} - \lambda\mathbf{I})^{-1}\mathbf{HC} & \mathbf{0}\end{bmatrix}$ must provide d_I rows that are linearly independent of all rows in $\begin{bmatrix}\mathbf{A} - \lambda\mathbf{I} & \mathbf{B}_I\end{bmatrix}$. We will derive conditions on the WCN topology to guarantee this.

Due to the assumption that the pair $(\mathbf{A}, \mathbf{C})$ is detectable, we have $\operatorname{rank}\left[\begin{smallmatrix}\mathbf{A} - \lambda\mathbf{I} \\ \mathbf{C}\end{smallmatrix}\right] = n$ for any unstable eigenvalue λ of $\mathbf{A}$ [30]. This means that for any set $I \subseteq \mathcal{M}$, there are at least d_I rows in the matrix $\begin{bmatrix}\mathbf{C} & \mathbf{0}\end{bmatrix}$ that are linearly independent of the rows in $\begin{bmatrix}\mathbf{A} - \lambda\mathbf{I} & \mathbf{B}_I\end{bmatrix}$. Let $J_1', J_2', \ldots, J_s'$ be all possible sets of d_I rows of $\mathbf{C}$ that satisfy this linear independence property, and let $\mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_s$ be the sets of d_I outputs of the plant corresponding to those rows. If we can guarantee that the row space of $\mathbf{C}_{J_i'}$ is contained in the row space of $\mathbf{E}_F(\mathbf{W} - \lambda\mathbf{I})^{-1}\mathbf{HC}$ for some i, then the right hand side of (15) will be at least $N + n$.

To satisfy this condition, we start by noting that $\mathbf{E}_F(\mathbf{W} - \lambda\mathbf{I})^{-1}\mathbf{H}$ in (15) is the transfer function of the WCN (where the outputs are taken to be nodes in the set $\mathcal{V}_{\mathcal{M}\setminus I}$) evaluated at λ. This matrix must have rank at least d_I in order for the right hand side of (15) to have rank $N + n$. To analyze this condition, we can consider a general *structured linear system* Σ. We are interested in the largest possible rank of the transfer function over all possible values of the nonzero free parameters and λ; this is called the *generic* rank of the transfer function matrix for the system. The following results relate this rank to a property of the graph associated with the system.

Lemma 2 ([31]): Let $\Sigma = (\mathbf{A}, \mathbf{B}, \mathbf{C})$ be a linear system, and let λ be such that $\mathbf{A} - \lambda\mathbf{I}$ is invertible. Then $\operatorname{rank}(\mathbf{M}(\lambda)) = \operatorname{rank}\left(\mathbf{C}(\mathbf{A} - \lambda\mathbf{I})^{-1}\mathbf{B}\right) + n$, where $\mathbf{M}(\lambda) = \left[\begin{smallmatrix}\mathbf{A} - \lambda\mathbf{I} & \mathbf{B} \\ \mathbf{C} & \mathbf{0}\end{smallmatrix}\right]$ □

Theorem 7 ([31]): Let $\Sigma = (\mathbf{A}, \mathbf{B}, \mathbf{C})$ be a structured linear system, and $\mathcal{G}_\Sigma$ its associated graph. The generic rank of the transfer function matrix is equal to the size of the largest linking from the input vertices to the output vertices in $\mathcal{G}_\Sigma$. □

We can now derive a condition that guarantees that the transfer function matrix has full rank when evaluated at certain values λ.

Lemma 3: Consider the structured system $\Sigma = (\mathbf{A}, \mathbf{B}, \mathbf{C})$ where the graph $\mathcal{G}_\Sigma$ contains a linking of size m from the input to the output vertices. Let $\mathcal{L} = \{\lambda_1, \lambda_2, \ldots, \lambda_r\}$ be a predefined finite set of nonzero complex numbers. Then,

$$\operatorname{rank}(\mathbf{C}(\mathbf{A} - \lambda_i\mathbf{I})^{-1}\mathbf{B}) = m, \; i \in \{1, 2, \ldots, r\} \qquad (16)$$

for almost any choice of free parameters in $(\mathbf{A}, \mathbf{B}, \mathbf{C})$. □

The proof of the lemma can be found in Appendix B.

Now that we have a handle on some rank properties of the matrix $\mathbf{E}_F(\mathbf{W} - \lambda\mathbf{I})^{-1}\mathbf{H}$, we return to the problem of

[8] For a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, given two subsets $\mathcal{V}_1, \mathcal{V}_2 \subset \mathcal{V}$, an *r-linking* from $\mathcal{V}_1$ to $\mathcal{V}_2$ is a set of r vertex disjoint paths, each with start vertex in $\mathcal{V}_1$ and end vertex in $\mathcal{V}_2$.

$$\begin{aligned}\operatorname{rank}\left(\tilde{\mathbf{M}}_{I,F}(\lambda)\right) &= \operatorname{rank}\begin{bmatrix}\mathbf{A}-\lambda\mathbf{I} & \mathbf{0} & \mathbf{B}_I\\ \mathbf{0} & \mathbf{W}-\lambda\mathbf{I} & \mathbf{0}\\ \mathbf{E}_F(\mathbf{W}-\lambda\mathbf{I})^{-1}\mathbf{HC} & \mathbf{0} & \mathbf{0}\end{bmatrix}\\ &= N + \operatorname{rank}\begin{bmatrix}\mathbf{A}-\lambda\mathbf{I} & \mathbf{B}_I\\ \mathbf{E}_F(\mathbf{W}-\lambda\mathbf{I})^{-1}\mathbf{HC} & \mathbf{0}\end{bmatrix}\end{aligned} \tag{15}$$

ensuring that the row space of $\mathbf{C}_{J_i'}$ is contained in the row space of $\mathbf{E}_F(\mathbf{W}-\lambda\mathbf{I})^{-1}\mathbf{HC}$, for some $i \in \{1,2,\ldots,s\}$. The following theorem provides topological conditions for the WCN to satisfy in order to guarantee that this condition holds.

Theorem 8: Consider the detectable and stabilizable (numerical) system $\Sigma = (\mathbf{A},\mathbf{B},\mathbf{C})$, along with a WCN. Let λ be an unstable eigenvalue of $\mathbf{A}$. For any subset $I \subseteq \mathcal{M}$, let $d_I = n - \operatorname{rank}\begin{bmatrix}\mathbf{A}-\lambda\mathbf{I} & \mathbf{B}_I\end{bmatrix}$. If for every possible subset I, there exists a subset J' of d_I plant outputs such that $\operatorname{rank}\begin{bmatrix}\mathbf{A}-\lambda\mathbf{I} & \mathbf{B}_I\\ \mathbf{C}_{J'} & \mathbf{0}\end{bmatrix} = n$, and the WCN contains a d_I linking from those outputs to $\mathcal{V}_{\mathcal{M}\setminus I}$, then for almost any choice of free parameters in $\mathbf{W}$ and $\mathbf{H}$, λ is not a fixed mode of the system $\tilde{\Sigma}$. Furthermore, if the above holds for every unstable eigenvalue of $\mathbf{A}$, then for almost any choice of parameters in $\mathbf{W}$ and $\mathbf{H}$ such that $\mathbf{W}$ is a stable matrix, system $\tilde{\Sigma}$ will have no unstable fixed modes. □

The proof of the theorem is provided in Appendix C.

To illustrate the use of the above theorem, we consider a WCN with the topology from Fig. 2, where the network provides a path between each sensor-actuator pair. Thus, if for any unstable eigenvalue λ the plant satisfies the condition that $\operatorname{rank}(\mathbf{A}-\lambda\mathbf{I}) = n-1$, then the WCN can guarantee closed-loop system stability. Note that the condition is true if all eigenvalues of $\mathbf{A}$ are distinct. However, even if some of the unstable eigenvalues are repeated (i.e., have algebraic multiplicity larger than one), the WCN can ensure system stability as long as the rank condition is satisfied. To specify this condition we can also use the notion of *geometric multiplicity* of eigenvalues of the plant: for any eigenvalue λ, $\operatorname{rank}(\mathbf{A}-\lambda\mathbf{I}) = n - d_\lambda$, where d_λ denotes its geometric multiplicity. Therefore, for the topology from Fig. 2, the WCN can stabilize all plants that have the maximal geometric multiplicity of all unstable eigenvalues (d) equal to 1. Similarly, we observe that the WCN from Fig. 4(b) can ensure stability of all plants with $d \leq 3$.

While the above result provides a method to check if the system has any fixed modes when controlled over a WCN, it requires all possible subsets of $\mathcal{M}$ to be tested. The following much simpler result provides a sufficient condition for the system to have no fixed modes.

Theorem 9: Consider the detectable and stabilizable system $\Sigma = (\mathbf{A},\mathbf{B},\mathbf{C})$, along with a WCN. Let d denote the largest geometric multiplicity of any unstable eigenvalue of $\mathbf{A}$. Suppose the vertex connectivity of the network is at least d, and each actuator has at least d WCN nodes in its neighborhood. Then, there exists a stabilizing WCN configuration. □

Proof: First, note that for any unstable eigenvalue λ of $\mathbf{A}$, we have $\operatorname{rank}(\mathbf{A}-\lambda\mathbf{I}) \geq n-d$, and thus the quantity d_I specified in Theorem 8 is no larger than d. Also, for any subset $I \subset \mathcal{M}$, let J' be the set of d_I outputs specified in that theorem, let $\mathcal{Y}'$ be the corresponding set of outputs, and let $\mathcal{V}_S'$ be the nodes in the WCN that receive information from the outputs in $\mathcal{Y}'$. Next, note that $|\mathcal{V}_{\mathcal{M}\setminus I}| \geq d \geq d_I$ by the assumption that each actuator has at least d wireless nodes in its neighborhood. Since the connectivity of the network is d, and since $|\mathcal{V}_{\mathcal{M}\setminus I}| \geq d$ and $|\mathcal{V}_S'| = |\mathcal{Y}'| = d_I \leq d$ (by the assumption from Remark 4), there exists a linking of size d_I from the set $\mathcal{V}_S'$ to $\mathcal{V}_{\mathcal{M}\setminus I}$ [32]. Thus, for almost any choice of parameters in $\mathbf{W}$ and $\mathbf{H}$ such that $\mathbf{W}$ is stable, all conditions in Theorem 8 are satisfied; the system will have no unstable fixed modes which means that it can be stabilized via a dynamic compensator at each actuator. ■

Remark 5: The linking and connectivity conditions from Theorems 8 and 9 are reminiscent of the classical requirement that a system having an unstable eigenvalue of geometric multiplicity d must have at least d outputs in order to be detectable [30]; they ensure that the new plant defined in (13) is detectable. Similar investigations of the sizes of cut-sets required for stabilization can be found in [33], [34], [35]. □

It is worth noting that the obtained results only ensure the existence of a stabilizing WCN configuration where each network node maintains a scalar state; these topological conditions do not provide any guarantees on the sizes of the states maintained by the actuators. As shown in [16], controllers used in the decentralized control setup from (10) could be (in the worst case) as large as the plant itself. As described before, in most industrial automation or process control scenarios this is not a concern since, due to physical constraints, actuators cannot typically be battery operated. This enables the use of more powerful computational platforms at the actuators, capable of implementing large-state controllers.

We have investigated this issue on several examples. In [36], we considered a setup where a single-input-single-output plant with three states is to be controlled using the WCN with two nodes, as in Fig. 2, where the additional link $v_2 \rightarrow v_1$ was added. We showed that for 3-state plants, stabilizing WCN configurations can be extracted where the single actuator maintains a state from $\mathbb{R}^2$. Furthermore, in [7] we showed that the same plant can be stabilized by a WCN consisting of nine nodes with a mesh topology, where the actuator maintains a dynamical controller with a scalar state. Finally, we generated stabilizing WCN configurations for 4×4 mesh networks used to control random plants with $n = 50$ states, $m = 10$ inputs, and $p = 10$ outputs, where all ten actuators maintain scalar states [7]. Consequently, it is natural to ask if there exists a dependency between the sizes of the controllers maintained at the actuators, plant dynamics, and the topology of the network. In the above examples, we were able to "shift" some of the computation from the actuators into the network, thus reducing the controllers' sizes. However, specifying a formal trade-off between the sizes of actuators' controllers and the network topology will be an avenue for future work.

VIII. Extensions to Point-To-Point Networks

Although we have focused thus far on dynamical system stabilization using a Wireless Control Network (which employs a local broadcast communication model), our analysis can be extended in a straightforward manner for control over networks with wired (or point-to-point) communication links. We consider the problem of network synthesis for the case where network coding over point-to-point communication links is used (as shown in Fig. 4(a)). Our goal is to provide topological conditions that guarantee that there exist linear dynamical controllers (at the actuators) that can stabilize the plant. We focus on two scenarios. We start with the case when the network delay (over each link in the network) is equal to the sampling period of the plant. We then investigate the case when an idealized, delay-free network is used. It is worth noting that this scenario can be used to model closed-loop systems where the speed of the network is much higher than the sampling period of the plant.

Suppose that $\mathcal{G}_c = (V_c, E_c \cup Y_{1:p} \cup U_{1:m})$ is a network with point-to-point links, where $Y_{1:p} = \cup_{i=1}^{p} Y_i$ represents the links coming into the network from the plant's sensors, and $U_{1:m} = \cup_{j=1}^{m} U_j$ represents the set of links coming out of the network into the plant's actuators. As is standard in linear network coding, the information sent on each outgoing edge from a given network node is a linear combination of information carried on the edges entering that node. Note that in the wired communication model, the linear combinations on each outgoing edge are allowed to be different. As shown in [12], from the graph $\mathcal{G}_c$ we can obtain the (unique) directed *labeled line graph* $\mathcal{B} = (\mathcal{V}_\mathcal{B}, \mathcal{E}_\mathcal{B})$, where $\mathcal{V}_\mathcal{B} = E_c \cup Y_{1:p} \cup U_{1:m}$, and for all $e_i, e_j \in \mathcal{V}_\mathcal{B}$, $(e_i, e_j) \in \mathcal{E}_\mathcal{B}$ if and only if there exist $v_1, v_2, v_3 \in V_c$ such that $e_i = (v_1, v_2)$ and $e_j = (v_2, v_3)$ (i.e., head(e_1) = tail(e_2)). Each link $(e_i, e_j) \in \mathcal{E}_\mathcal{B}$ is labeled with the coefficient (i.e., weight) assigned to the information received over edge e_i in the linear combination that is used to produce information over e_j. An illustration of this procedure is shown in Fig. 4(b), where the labeled line graph is given for the network from Fig. 4(a). Note that each link in the initial graph corresponds to a unique vertex in the labeled line graph.

If each link in the initial network introduces a fixed communication delay (as in Time-Triggered networks [6]), the labeled line graph directly corresponds to the WCN model. In this case the matrices $\mathbf{W}, \mathbf{H}, \mathbf{G}$ contain the gains between network links, between inputs and network links, and network links and the outputs, respectively. Therefore, if we are able to derive a stabilizing configuration for the corresponding WCN, the same configuration (i.e., the network coding parameters and parameters of the controllers) would guarantee stability when network coding is used in the initial point-to-point network $\mathcal{G}_c$. We start by noting that Theorems 8 and 9 specify sufficient conditions for the WCN topology to ensure that such a configuration exists. These conditions require a sufficient vertex cut (i.e., linking) for the WCN topology. Since each vertex in the WCN corresponds to a specific edge in the initial network (and *vice versa*), we can directly obtain sufficient topological conditions for a network that uses network coding over point-to-point links. Thus, we can specify a theorem equivalent to Theorem 9 (a theorem equivalent to Theorem 8 can also be stated).

Theorem 10: Consider the detectable and stabilizable system $\Sigma = (\mathbf{A}, \mathbf{B}, \mathbf{C})$, and a network whose link communication delay is equal to the plant's sampling time and which employs network coding over point-to-point links. Let d denote the largest geometric multiplicity of any unstable eigenvalue of $\mathbf{A}$. If the minimal edge cut of the network between sensors and actuators is at least d, then the system $\tilde{\Sigma}$ can be stabilized via a dynamic compensator at each actuator. □

Similar results can be obtained in the case with delay-free communication networks, where the information injected in the network by the plant's sensors is expected to be instantaneously available at the actuators. In this case, as described in [12], for the directed labeled graph of the initial network we can define $\mathbf{W}$ – the adjacency matrix of the labeled graph. Here, w_{ij} is the weight assigned to the edge e_i in the linear combination used to derive e_j (if head$(e_i) \neq$ tail(e_j) then $w_{ij} = 0$).[9] Using the matrix $\mathbf{W}$, as in [12] it can be shown that for any set $I \subseteq \mathcal{M}$, $\mathbf{E}_F(\mathbf{W} - \mathbf{I})^{-1}\mathbf{H}$ is the transfer matrix of the network, from the input edges (i.e., from the sensors) to the output edges (i.e., to the actuators specified in the set $\mathcal{A}_{\mathcal{M}\setminus I} = \bigcup_{i \in \mathcal{M}\setminus I} a_i$).[10] This is equal to the WCN transfer function, evaluated at $\lambda = 1$, which is used in the proof of Theorem 8. Therefore, by using the same approach from the proof of Theorem 8, we can formulate theorems equivalent to Theorems 8 and 9 (as in the case where networks introduce delay). This means that, even for delay-free networks that use network coding over point-to-point links, Theorem 10 specifies sufficient conditions for the existence of network coding parameters for which the plant can be stabilized via controllers at the actuators.

As an illustration, we consider the networks from Fig. 2 and Fig. 4(b). In the first case, all plants with the maximal geometric multiplicity of all unstable eigenvalues (d) equal to 1 can be stabilized with controllers at actuators. Similarly, for the network from Fig. 4(a) and for all plants with $d \leq 3$ there exist network coding parameters and stabilizing controllers at the actuators.

IX. Conclusion

In this paper, we have studied the problem of stabilizing a given dynamical system over a network. In contrast to traditional approaches that treat the network purely as a routing mechanism (delivering sensor measurements to controllers, and control inputs to actuators), we propose a fundamentally different approach that relies on inducing carefully chosen *dynamics* on the network (via the form of a simple distributed algorithm), and using those dynamics to stabilize the plant. This approach does away with end-to-end routing entirely, and only requires that nodes transmit information to their nearest neighbors at each time-step. We provided topological conditions on the network that allow the system to be stabilized

[9] Note that in this case, the initial graph has to be acyclic, which in-turn causes the line graph to be acyclic.

[10] In this context we can also observe that the result from Theorem 7 is a structural equivalent for the results from [12], [25], [35] that relate the size of the minimal edge cut of the network with the rank of the transfer matrix.

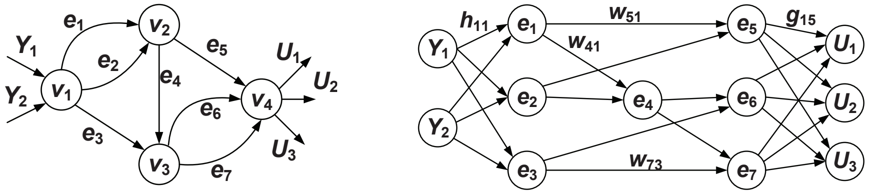


Fig. 4. (a) Point to point communication in a simple network [12]; Sources $\mathbf{Y}_{1:2}$ represent input processes, $\mathbf{U}_{1:3}$ denotes the network outputs; (b) The directed labeled line graph for the graph from (a) (only some of the links have been labeled to reduce clutter).

in this manner. Specifically, we showed that if the network is sufficiently well connected, each node and actuator can use a *linear iterative strategy* with appropriately chosen weights to stabilize the plant; furthermore, the connectivity required is determined by the *dynamics* of the plant, rather than the number of source nodes (as in traditional information transmission scenarios). Our approach also extends in a straightforward manner to wired (point-to-point) networks via a standard graph transformation.

Appendix A
Proof of Theorem 6

Proof: A directed graph $\mathcal{G}_\Sigma = \{\mathcal{V}_\Sigma, \mathcal{E}_\Sigma\}$, representing the structured system Σ, can be uniquely decomposed into k strongly connected components $\xi = \{\xi_1, ..., \xi_k\}$. A component ξ_i is referred to as a root component if no vertex in the component has incoming edges from vertices in any other component. Also, ξ_j is called a leaf component if no vertex in ξ_j has an outgoing edge to a vertex in any other component.

Consider a directed acyclic graph $\mathcal{G}_\xi = \{\xi \cup \mathcal{U}_I \cup \mathcal{Y}_J, \mathcal{E}_\xi \cup \mathcal{E}_{\mathcal{I}\xi} \cup \mathcal{E}_{\mathcal{J}\xi}\}$, where $(\xi_i, \xi_j) \in \mathcal{E}_\xi$ if and only if component ξ_j has an incoming edge from a vertex in ξ_i, and

$$\mathcal{E}_{\mathcal{I}\xi} = \left\{ (u_i, \xi_t) \;\middle|\; \begin{array}{c} i \in I, \xi_t \text{ is a root component from } \xi, \\ \xi_t \text{ has an edge from input vertex } u_i \end{array} \right\},$$

$$\mathcal{E}_{J\xi} = \left\{ (\xi_t, y_j) \;\middle|\; \begin{array}{c} j \in J, \xi_t \text{ is a leaf component in } \xi, \\ \text{output vertex } y_j \text{ has an edge from } \xi_t \end{array} \right\}.$$

The graph $\mathcal{G}_\xi$ is called a *condensation* of the initial graph [18].

Since the system Σ is structurally stabilizable and detectable, each leaf component has to be connected to an output vertex $y_j \in \mathcal{Y}_J$ and each root component is connected to an input vertex $u_i \in \mathcal{U}_I$. We now use Algorithm 1 to introduce $\mathcal{E}_\mathcal{F}$, a set of feedback links between output vertices from $\mathcal{Y}_J$ and input vertices from $\mathcal{U}_I$.

In step **1** there has to exist an output y_{j_1} as components connected to u_{i_1} have to be connected to at least one output (since the system is detectable). Step **2** will create a cycle $\mathcal{C}$ in the newly obtained graph $\mathcal{G}_{\xi,\mathcal{F}} = \{\xi \cup \mathcal{U}_I \cup \mathcal{Y}_J, \mathcal{E}_\xi \cup \mathcal{E}_{\mathcal{I}\xi} \cup \mathcal{E}_{\mathcal{J}\xi} \cup \mathcal{E}_\mathcal{F}\}$ that contains the same number of input and output nodes. In step **3**, pairs of input and output vertices are selected from all input and output vertices from I and J that are not a part of the cycle. If (y_j, u_i) is such a pair, y_j is not reachable from u_i in the initial graph $\mathcal{G}_\xi$ (otherwise vertex u_i would be selected in step **2**). In addition, there has to exist a vertex $u_r \in \mathcal{C}$ from which vertex y_j can be reached, since if that is not the case the vertices u_r, y_j would be selected in step **2**. Similarly, there exists a vertex $y_l \in \mathcal{C}$ reachable from u_i. Therefore, in the newly created graph $\mathcal{G}_{\xi,\mathcal{F}}$ there would exist a cycle containing vertices y_j, u_i, y_l, u_r.

```
Algorithm 1 Creating a minimal set of feedback connections
1. Select an input vertex $u_{i_1} \in \mathcal{U}_I$ and a corresponding output vertex $y_{j_1} \in \mathcal{Y}_J$ such that $y_{j_1}$ is reachable from $u_{i_1}$ in the graph $\mathcal{G}_\xi$.
2. At iteration $t \geq 1$, select an input vertex $u_{i_{t+1}} \in \mathcal{U}_I \setminus \{u_{i_1}, ..., u_{i_t}\}$ such that there exists an output vertex $y_{j_{t+1}} \in \mathcal{Y}_J \setminus \{y_{j_1}, ..., y_{j_t}\}$ reachable from $u_{i_{t+1}}$ in the graph $\mathcal{G}_\xi$. If such an input $u_{i_{t+1}}$ does not exist, add the edge $(y_{j_t}, u_{i_1})$ to $\mathcal{E}_\mathcal{F}$, and go to the next step. Otherwise, add the edge $(y_{j_t}, u_{i_{t+1}})$ to the set $\mathcal{E}_\mathcal{F}$, set $t \leftarrow t+1$ and repeat step 2.
3. If $\{u_{i_1}, ..., u_{i_t}\} \neq I$ and $\{y_{j_1}, ..., y_{j_t}\} \neq J$ then select $u_{i_{t+1}} \notin \{u_{i_1}, ..., u_{i_t}\}$ and $y_{j_{t+1}} \notin \{y_{j_1}, ..., y_{j_t}\}$ and add the edge $(y_{j_{t+1}}, u_{i_{t+1}})$ to $\mathcal{E}_\mathcal{F}$. Set $t \leftarrow t+1$ and repeat step 3.
4. If $\{u_{i_1}, ..., u_{i_t}\} = I$ then for all $y_j \notin \{y_{j_1}, ..., y_{j_t}\}$ add the edge $(y_j, u_i)$ to $\mathcal{E}_\mathcal{F}$, where $u_i$ is an input vertex from which $y_j$ can be reached in the initial graph $\mathcal{G}_\xi$.
5. If $\{y_{j_1}, ..., y_{j_t}\} = J$ then for all $u_i \notin \{u_{i_1}, ..., u_{i_t}\}$ add the edge $(y_j, u_i)$ to $\mathcal{E}_\mathcal{F}$, where $y_j$ is an output vertex reachable from $u_i$ in the graph $\mathcal{G}_\xi$.
```

After step **3**, $\min(|I|, |J|)$ feedback links will be added to the set $\mathcal{E}_\mathcal{F}$. Finally, in steps **4** and **5** the remaining output or input vertices, respectively, will be connected to the vertices from which they can be reached. Hence, Algorithm 1 will use $\max(|I|, |J|)$ feedback connections, and for each input and output vertex $u_i \in I$ and $y_j \in J$ such that y_j is reachable from u_i in the graph $\mathcal{G}_\xi$, there will exist a path from y_j to u_i in the new graph $\mathcal{G}_{\xi,\mathcal{F}}$ that contains an edge from $\mathcal{E}_\mathcal{F}$. Thus, each component ξ_i will belong to a strongly connected component with an edge from $\mathcal{E}_\mathcal{F}$, which (from Corollary 1) implies that the system will not have structural fixed modes. ■

Appendix B
Proof of Lemma 3

Proof: We first show that there exist free parameters for which $\text{rank}(\mathbf{C}(\mathbf{A} - \lambda_i\mathbf{I})^{-1}\mathbf{B}) = m$ for $i \in \{1, 2, ..., r\}$. We then show that this holds for almost any choice of free parameters.

If the graph $\mathcal{G}_\Sigma$ contains an m-linking, Theorem 7 and Lemma 2 tell us that there is a numerical choice of free parameters and λ for which $\text{rank}(\mathbf{M}(\lambda)) = n + m$. Thus, there must exist an $(n+m)$-th order minor of $\mathbf{M}(\lambda)$ that is nonzero. If we replace λ with a variable z, and revert all of the nonzero values in the system matrices to free parameters, this minor is a nonzero polynomial $f(z)$ in z and the free parameters (we leave out the free parameters in the argument of $f(\cdot)$ for clarity).

For the specific choice of free parameters that guarantees that $\text{rank}(\mathbf{C}(\mathbf{A}-\lambda\mathbf{I})^{-1}\mathbf{B}) = m$, if $f(z)$ has no roots in common with the set $\mathcal{L}$, then we have shown that there exists one choice of free parameters for which (16) holds (because $f(\lambda_i) \neq 0$ for $i \in \{1, 2, \ldots, r\}$, which means that $\text{rank}(\mathbf{M}(\lambda_i)) = n + m$). Otherwise let λ_{min} be the nonzero root of $f(z)$ with smallest magnitude, and let α be a positive real number such that $\alpha\lambda_{min}$ has larger magnitude than the largest element of $\mathcal{L}$. Then, if we scale $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ by α, one can verify that the resulting $(n+m)$-th order minor of $\mathbf{M}(z)$ becomes $\alpha^{n+m}f(\frac{z}{\alpha})$. The roots of this polynomial are the roots of $f(z)$ scaled by α, and so all nonzero roots will have magnitude larger than any elements in $\mathcal{L}$. Thus, there exists a choice of free parameters for which (16) holds.

To show that (16) holds for almost any choice of free parameters, we denote with $g(z)$ a polynomial whose roots are the elements of $\mathcal{L}$ (extended to include complex conjugate roots if necessary). By the above argument, $f(z)$ and $g(z)$ will have no roots in common for some choice of free parameters, or equivalently, the *resultant* of $f(z)$ and $g(z)$ will be nonzero.[11] If we revert $f(z)$ to be a polynomial in the free parameters, the resultant of $g(z)$ and $f(z)$ will also be a nonzero polynomial in the free parameters. The set of parameters causing this resultant to be zero are the parameters for which $f(z)$ has a root in $\mathcal{L}$. Thus, the set of free parameters for which (16) does not hold lies on an algebraic variety, which proves the lemma. ■

Appendix C
Proof of Theorem 8

Proof: For a given subset I and the corresponding set $\mathcal{V}_{\mathcal{M}\setminus I}$, denote the graph of the structured system $\Sigma_{WCN} = (\mathbf{W}, \mathbf{H}, \mathbf{E}_F)$ by $\mathcal{G}_{\Sigma_{WCN}}$.[12] Noting that the inputs to the WCN are the outputs of the plant, the input vertices in $\mathcal{G}_{\Sigma_{WCN}}$ are given by $\mathcal{Y}$. Furthermore, denote the output vertices of $\mathcal{G}_{\Sigma_{WCN}}$ by $\mathcal{V}_{\mathcal{M}\setminus I}$. Consider any subset $I \subseteq \mathcal{M}$ for which $d_I > 0$, and let $\mathcal{Y}'$ be the set of d_I outputs corresponding to the set J' described in the theorem. According to the assumption in the theorem, the graph $\mathcal{G}_{\Sigma_{WCN}}$ contains a linking of size d_I from these outputs to $\mathcal{V}_{\mathcal{M}\setminus I}$. Let $\mathbf{H}_{J'}$ denote the matrix consisting of the columns of $\mathbf{H}$ corresponding to the outputs in set $\mathcal{Y}'$, and consider the system $(\mathbf{W}, \mathbf{H}_{J'}, \mathbf{E}_F)$. The graph of this system is obtained simply by removing the vertices that are not in $\mathcal{Y}'$ from the graph $\mathcal{G}_{\Sigma_{WCN}}$. Since this reduced graph has a d_I-linking from the inputs to the outputs, Theorem 7 and Lemma 3 indicate that $\mathbf{E}_F(\mathbf{W}-\lambda\mathbf{I})^{-1}\mathbf{H}_{J'}$ will have rank d_I for almost choice of free parameters in $\mathbf{W}$ and $\mathbf{H}_{J'}$. Thus, $\mathbf{E}_F(\mathbf{W}-\lambda\mathbf{I})^{-1}\mathbf{H}_{J'}\mathbf{C}_{J'}$ will have rank d_I, and

$$\text{rank}\begin{bmatrix} \mathbf{A}-\lambda\mathbf{I} & \mathbf{B}_I \\ \mathbf{E}_F(\mathbf{W}-\lambda\mathbf{I})^{-1}\mathbf{H}_{J'}\mathbf{C}_{J'} & \mathbf{0} \end{bmatrix} = n.$$

The matrix $\begin{bmatrix} \mathbf{A}-\lambda\mathbf{I} & \mathbf{B}_I \\ \mathbf{E}_F(\mathbf{W}-\lambda\mathbf{I})^{-1}\mathbf{HC} & \mathbf{0} \end{bmatrix}$ has rank n for some choice of $\mathbf{W}$ and $\mathbf{H}$ (i.e., by setting the columns of $\mathbf{H}$ that are not in J' to be zero, and choosing all other parameters almost arbitrarily). Therefore, there is an n-th order minor of the above matrix that is nonzero. Setting all nonzero entries in the columns of $\mathbf{H}$ that are not in $\mathbf{H}_{J'}$ to be free parameters, this minor will be a nonzero polynomial in those parameters. Thus, the set of parameters for which the rank of the above matrix is less than n lies on an algebraic variety, and so the matrix has rank at least n for almost any choice of free parameters.

The above analysis holds for every subset $I \subseteq \mathcal{M}$, and thus λ is not a fixed mode of the system. Furthermore, if the conditions in the theorem hold for every unstable eigenvalue of $\mathbf{A}$, all of these eigenvalues will not be fixed modes for almost any choice of free parameters in $\mathbf{W}$ and $\mathbf{H}$. Finally, note that the eigenvalues of $\tilde{\mathbf{A}}$ in (13) are the union of the eigenvalues of $\mathbf{A}$ and $\mathbf{W}$. The set of free parameters that makes $\mathbf{W}$ a stable matrix has measure greater than zero, and the above analysis will hold for almost any such choice of parameters. This guarantees that no eigenvalue of $\mathbf{W}$ can be an unstable fixed mode, which concludes the proof. ■

References

[1] M. Pajic, S. Sundaram, G. J. Pappas, and R. Mangharam, "Topological Conditions for Wireless Control Networks," in *Proc. 50th IEEE Conference on Decision and Control*, 2011, pp. 2353–2360.

[2] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proc. IEEE*, vol. 95, no. 1, pp. 138–162, 2007.

[3] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, "Foundations of control and estimation over lossy networks," *Proc. IEEE*, vol. 95, pp. 163–187, 2007.

[4] C. L. Robinson and P. R. Kumar, "Optimizing controller location in networked control systems with packet drops," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 4, pp. 661–671, 2008.

[5] M. Pajic and R. Mangharam, "Embedded virtual machines for robust wireless control and actuation," in *RTAS'10: 16th IEEE Real-Time and Embedded Technology and Applications Symposium*, 2010, pp. 79–88.

[6] R. Alur, A.D'Innocenzo, K. H. Johansson, G. J. Pappas, and G. Weiss, "Compositional modeling and analysis of multi-hop control networks," *IEEE Trans. Autom. Control*, vol. 56, no. 10, pp. 2345–2357, 2011.

[7] M. Pajic, S. Sundaram, R. Mangharam, and G. J. Pappas, "The Wireless Control Network: A New Approach for Control over Networks," *IEEE Trans. Autom. Control*, vol. 56, no. 10, pp. 2305–2318, 2011.

[8] M. Pajic, S. Sundaram, J. Le Ny, G. J. Pappas, and R. Mangharam, "Closing the loop: A simple distributed method for control over wireless networks," in *Proc. 11th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2012, pp. 25–36.

[9] A. Giridhar and P. Kumar, "Computing and communicating functions over sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 755 – 764, 2005.

[10] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[11] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation and consensus using linear iterative strategies," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 4, pp. 650–660, 2008.

[12] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782 – 795, 2003.

[13] S. Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 3, pp. 371 –381, 2003.

[14] S.-Y. Li, Q. Sun, and Z. Shao, "Linear network coding: Theory and algorithms," *Proc. IEEE*, vol. 99, no. 3, pp. 372 –387, 2011.

[15] P. Elias, A. Feinstein, and C. Shannon, "A note on the maximum flow through a network," *IEEE Trans. Inf. Theory*, vol. 2, no. 4, pp. 117 – 119, 1956.

[16] S.-H. Wang and E. Davison, "On the stabilization of decentralized control systems," *IEEE Trans. Autom. Control*, vol. 18, no. 5, pp. 473–478, 1973.

[17] M. Sezer and D. Siljak, "Structurally fixed modes," *Systems & Control Letters*, vol. 1, no. 1, pp. 60–64, 1981.

[11] The resultant of two polynomials is the determinant of the Sylvester matrix associated with those polynomials, and is nonzero if and only if the polynomials have no roots in common [37].

[12] Although $\mathbf{E}_F$ is a zero-one matrix, as in Remark 4 the fact that the matrix contains a single 1 in each row allows us to consider the system as structured.

[18] D. Siljak, *Decentralized control of complex systems*. Academic Press, Inc, 1991.

[19] J. Lavaei and S. Sojoudi, "Time complexity of decentralized fixed-mode verification," *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 971–976, 2010.

[20] J. L. Willems, "Time-varying feedback for the stabilization of fixed modes in decentralized control systems," *Automatica*, vol. 25, no. 1, pp. 127–131, 1989.

[21] P. Khargonekar and A. Ozguler, "Decentralized control and periodic feedback," *IEEE Trans. Autom. Control*, vol. 39, no. 4, pp. 877 –882, 1994.

[22] Z. Gong and M. Aldeen, "Stabilization of decentralized control systems," *Journal of Mathematical Systems Estimation and Control*, vol. 7, pp. 111–114, 1997.

[23] J.-M. Dion, C. Commault, and J. van der Woude, "Generic properties and control of linear structured systems: a survey," *Automatica*, vol. 39, no. 7, pp. 1125–1144, 2003.

[24] V. Pichai, M. E. Sezer, and D. D. Siljak, "A graph-theoretic characterization of structurally fixed modes," *Automatica*, vol. 20, no. 2, pp. 247–250, 1984.

[25] S. Y. Park and A. Sahai, "An algebraic mincut-maxflow theorem," in *2011 IEEE International Symposium on Information Theory Proceedings (ISIT)*, 2011, pp. 608–612.

[26] L. El Ghaoui, F. Oustry, and M. Ait Rami, "A cone complementarity linearization algorithm for static output-feedback and related problems," *IEEE Trans. Autom. Control*, vol. 42, no. 8, pp. 1171–1176, 1997.

[27] J. Han and R. Skelton, "An LMI optimization approach for structured linear controllers," in *Proc. 42nd IEEE Conference on Decision and Control*, 2003, pp. 5143–5148.

[28] B. D. Anderson and D. J. Clements, "Algebraic characterization of fixed modes in decentralized control," *Automatica*, vol. 17, no. 5, pp. 703–712, 1981.

[29] M. E. Sezer, "Minimal essential feedback patterns for pole assignment using dynamic compensation," in *Proc. 22nd IEEE Conference on Decision and Control*, 1983, pp. 28–32.

[30] C. Chen, *Linear Systems, Theory and Design*. Oxford University Press, 1999.

[31] J. W. van der Woude, "A graph-theoretic characterization for the rank of the transfer matrix of a structured system," *Mathematics of Control, Signals and Systems*, vol. 4, no. 1, pp. 33–40, 1991.

[32] D. B. West, *Introduction to Graph Theory*. Prentice-Hall Inc., Upper Saddle River, New Jersey, 2001.

[33] A. Matveev and A. Savkin, "Multirate stabilization of linear multiple sensor systems via limited capacity communication channels," *SIAM journal on control and optimization*, vol. 44, no. 2, pp. 584–617, 2005.

[34] S. Yuksel, "Decentralized computation and communication in stabilization of distributed control systems," in *Proc. Information Theory and Applications Workshop, ITA*, vol. 9, 2009.

[35] S. Y. Park and A. Sahai, "Network coding meets decentralized control: Capacity-stabilizabililty equivalence," in *Proc. 50th IEEE Conference on Decision and Control*, 2011, pp. 4817–4822.

[36] M. Pajic, S. Sundaram, R. Mangharam, and G. J. Pappas, "The wireless control network: Topological conditions for system stabilization," University of Pennsylvania, Tech. Rep., 2011.

[37] G. Bartolini, A. Ferrara, and E. Usani, "Chattering avoidance by second-order sliding mode control," *IEEE Trans. Autom. Control*, vol. 43, no. 2, pp. 241 –246, 1998.

Miroslav Pajic (S'06) received the Dipl. Ing. and M.S. degrees in electrical engineering from the University of Belgrade, Serbia, in 2003 and 2007, respectively, and the M.S. and Ph.D. degrees in electrical engineering from the University of Pennsylvania, Philadelphia, in 2010 and 2012, respectively.

He is a Postdoctoral Fellow in the Department of Electrical & Systems Engineering at the University of Pennsylvania. His research interests include cyber-physical systems, embedded and distributed/networked control systems, real-time embedded systems, and high-confidence medical device systems.

Dr. Pajic received several awards including the 2011 ACM SIGBED Frank Anger Memorial Award, and the Best Student Paper award at the 2012 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS).

Rahul Mangharam (M'02) received the B.S., M.S., and Ph.D. degrees in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 2000, 2002, and 2008 respectively.

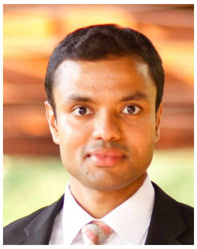
He is the Stephen J Angello Chair and Assistant Professor in the Dept. of Electrical & Systems Engineering and Dept. of Computer & Information Science at the University of Pennsylvania. He is the Director of the Real-Time and Embedded Systems Lab. His current interests are in real-time scheduling and control algorithms for networked embedded systems with applications in automotive systems, medical devices, energy-efficient buildings and wireless control networks.

Dr. Mangharam received the 2012 Intel Early Faculty Career Award and was selected by the National Academy of Engineering for the 2012 US Frontiers of Engineering.

George J. Pappas (S'90-M'91-SM'04-F'09) received the Ph.D. degree in electrical engineering and computer sciences from the University of California, Berkeley (where he received the Eliahu Jury Award for Excellence in Systems Research), in 1998.

He is currently the Joseph Moore Professor of Electrical and Systems Engineering at the University of Pennsylvania, Philadelphia. He is a member of the General Robotics, Automation, Sensing and Perception (GRASP) Laboratory and serves as the Chair of the Department of Electrical and Systems Engineering. His current research interests include hybrid and embedded systems, hierarchical control systems, distributed control systems, nonlinear control systems, with applications to robotics, unmanned aerial vehicles, biomolecular networks, and green buildings.

Dr. Pappas has received numerous awards, including the National Science Foundation (NSF) CAREER Award in 2002, the NSF Presidential Early Career Award for Scientists and Engineers in 2002, the 2009 George S. Axelby Outstanding Paper Award, and the 2010 Antonio Ruberti Outstanding Young Researcher Prize.



Shreyas Sundaram (M'09) is an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Waterloo. He received his MS and PhD degrees in electrical engineering from the University of Illinois at Urbana-Champaign in 2005 and 2009, respectively. He was a Postdoctoral Researcher in the GRASP Laboratory at the University of Pennsylvania from 2009-2010. His research interests include the analysis of dynamics on networks, secure and fault-tolerant control, network science, and the application of graph theory to systems analysis.

Dr. Sundaram received the M. E. Van Valkenburg Graduate Research Award and the Robert T. Chien Memorial Award from the University of Illinois, both for excellence in research. He was a finalist for the Best Student Paper Award at the 2007 and 2008 American Control Conferences.