Networked Realization of Discrete-Time Controllers

Fei Miao   Miroslav Pajic   Rahul Mangharam   George J. Pappas

Abstract— We study the problem of mapping discrete-time linear controllers into potentially higher order linear controllers with predefined structural constraints. Our work has been motivated by the Wireless Control Network (WCN) architecture, where the network itself behaves as a distributed, structured dynamical compensator. We make connections to model reduction theory to derive a method for the controller embedding based on minimization of the H_∞-norm of the error system. This allows us to frame the problem as synthesis of optimal structured linear controllers, which enables the utilization of design-time iterative procedures for systems' approximation. Finally, we illustrate the use of the mapping procedure by embedding PID controllers into the WCN substrate, and show how to reduce the computation overhead of the approximation procedure.

I. Introduction

In this paper, we address the problem of embedding discrete-time linear controllers into structured computational substrate – i.e., potentially higher order linear controllers with structural constraints. This work has been motivated by a recently introduced distributed control scheme from [1]. Unlike traditional networked control schemes that use the standard network architecture "sensor → channel → controller/estimator → channel → actuator", the Wireless Control Network (WCN) employs a fully distributed control scheme where the entire network itself acts as a controller.

Each node in a WCN behaves as a dynamical controller instead of routing information to and from the controller; it maintains a state, and at each time-step the state value is updated to be a linear combination of the node's previous state and the states of the neighboring nodes and sensors. As shown in [1], this scheme causes the entire network to behave as a *structured linear compensator*, with sparsity constraints imposed by the underlaying network topology. Besides introducing a very small overhead, making it suitable for implementation on resource constrained wireless nodes, the proposed scheme has several additional benefits (see Section II). However, to accelerate proliferation of this technology, it is necessary to provide a method to map existing controllers, like PID controller or the ones developed using the standard networked control system techniques (e.g., [2], [3], [4]), into the WCN computational substrate.

The exact realization of linear controllers using a system of networked, spatially interconnected systems has drawn a lot of attention in the past decade (e.g., [5], [6], [7], [8]). However, the main focus has been on specific networked

topologies (i.e., structural constraints) or topological conditions for which the controller can be implemented on the provided structured computational substrate. In this work, we consider the general case of structural constraints, which usually means that it is not possible to exactly match the initial and derived controllers. Consequently, instead of the exact mapping, we focus on the approximation of linear systems with potentially higher order, structurally constrained systems. This problem effectively presents a dual of the standard model reduction problem – approximating a high-order system by a lower-order one (according to certain criteria). Thus, similarly to [9], we employ model reduction techniques to specify an error system using the H_∞-norm. Note that the H_∞-norm of the difference of the two systems has been a widely used measure of the model reduction error, receiving considerable attention in the literature (e.g., see survey [10]). Grigoriadis derived necessary and sufficient conditions for the existence of a solution to the H_∞ model reduction problems [11].

The specified error system allows for the problem formulation as synthesis of an optimal structured linear controller (SLC). The difference is that our structured controller (e.g., the WCN) has some freedom in the system dimension to compensate for the structural constraints – i.e., to reduce the approximation error we can increase the sizes of states maintained by some nodes in the network. The SLC design problem was addressed in [12], [13], [14], where iterative procedures were used to solve the corresponding optimization problems formulated using Linear Matrix Inequalities (LMIs). We apply these algorithms to obtain the error system with the local minimal $\mathbf{H}_\infty$ norm. Furthermore, to reduce the approximation error, we provide a computationally more efficient method to expand the derived structured controller by increasing its size (while maintaining the structure).

This paper is organized as follows. In Section II, we present an overview of the WCN. In Section III, we formulate the structural mapping problem as an H_∞-based optimization problem, before we present (in Section IV) an iterative method that can be used to solve it. Section V illustrates the mapping of PID controllers into the WCN and provides an approach to reduce the computation overhead of the mapping procedure. Finally, Section VI provides concluding remarks.

1) Notation: Consider a standard linear time-invariant (LTI) system $\Sigma = (\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$

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

with $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times m}$, $\mathbf{C} \in \mathbb{R}^{p \times n}$ and $\mathbf{D} \in \mathbb{R}^{p \times m}$. We denote the structural constraints of the system by

$$\Psi_\Sigma = \Psi_{\mathbf{A}} \times \Psi_{\mathbf{B}} \times \Psi_{\mathbf{C}} \times \Psi_{\mathbf{D}}, \qquad (2)$$

where $\Psi_{\mathbf{A}}, \Psi_{\mathbf{B}}, \Psi_{\mathbf{C}}, \Psi_{\mathbf{D}}$ specify the constraints on the matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and $\mathbf{D}$, respectively. For example, we say that $\mathbf{A} \in \Psi_{\mathbf{A}} \subseteq \mathbb{R}^{n \times n}$ satisfies the structural constraints when $[\mathbf{A}]_{i,j} = 0$ if, for any k, the value of $\mathbf{x}_j[k]$ does not affect $\mathbf{x}_i[k+1]$. Consequently, other matrix elements are free parameters.

II. MOTIVATION - WIRELESS CONTROL NETWORKS

We consider the setup presented in Figure 1, where an LTI plant is to be controlled by a multi-hop wireless network. The network is described by a graph $\mathcal{G}\{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{V} = \{v_1, \ldots, v_N\}$ denotes the set of nodes, while $\mathcal{E}$ is the edge set. To model the WCN, let $z_i[k]$ denote node v_i's state at time step k. The linear update procedure can be described as

$$z_i[k+1] = w_{ii}z_i[k] + \sum_{v_j \in \mathcal{N}_{v_i}} w_{ij}z_j[k] + \sum_{s_j \in \mathcal{N}_{v_i}} h_{ij}y_j[k], \quad (3)$$

where $\mathcal{N}_{v_i}$ denotes the neighborhood of the node v_i – i.e., the set of nodes and sensors whose transmissions can be received by v_i.

Furthermore, to control the plant, each control input $u_i[k]$ is derived as a linear combination of the states from the nodes in the neighborhood $\mathcal{N}_{a_i}$ of the actuator a_i (that controls the input):

$$u_i[k] = \sum_{j \in \mathcal{N}_{a_i}} g_{ij}z_j[k]. \qquad (4)$$

Therefore, the behavior of the WCN is specified by the link weights w_{ij}, h_{ij} and g_{ij}, which define the linear combinations computed by the nodes and actuators in the network. If the states of all nodes at time step k are aggregated into the state vector $\mathbf{z}[k] = \begin{bmatrix} z_1[k] & z_2[k] & \ldots & z_N[k] \end{bmatrix}^T$, the behavior of the entire network can be modeled as:

$$\begin{aligned} \mathbf{z}[k+1] &= \mathbf{W}\mathbf{z}[k] + \mathbf{H}\mathbf{y}[k] \ , \\ \mathbf{u}[k] &= \mathbf{G}\mathbf{z}[k] \end{aligned} \qquad (5)$$

for all $k \in \mathbb{N}$. It is important to note here that in the above model $\mathbf{W} \in \mathbb{R}^{N \times N}, \mathbf{H} \in \mathbb{R}^{N \times p}, \mathbf{G} \in \mathbb{R}^{m \times N}$, and 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 $v_j \notin \mathcal{N}_{a_i}$. Thus, the matrices $\mathbf{W}, \mathbf{H}$ and $\mathbf{G}$ are *structured*, with sparsity constraints imposed by the underlying network topology. Throughout the rest of the paper, we will also use Ψ as in (2) to denote the set of all tuples $(\mathbf{W}, \mathbf{H}, \mathbf{G}) \in \mathbb{R}^{N \times N} \times \mathbb{R}^{N \times p} \times \mathbb{R}^{m \times N}$ that satisfy the aforementioned sparsity constraints. Consequently, the WCN scheme causes the network to act as a *structured dynamical compensator*.

The WCN scheme has several benefits; it imposes low computation and communication overhead to the nodes in the network, allows easy scheduling of wireless transmissions, and enables *compositional design* (that effectively decouples the design and analysis of several control loops implemented on the same network). In [1], [15], WCN synthesis procedures were proposed, which ensure system stability/optimality for a provided network topology. On the
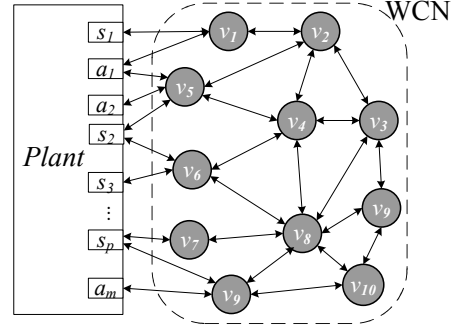

Fig. 1: A multi-hop Wireless Control Network, where the network *itself* acts as a distributed controller [15].

other hand, in [16] a set of topological conditions for the existence of a stabilizing WCN configuration was derived.[1]

In the above model of the WCN we assumed a scenario where each node maintains a scalar state. As shown in [1], the more general case, where each node can maintain a vector state with possibly different dimensions, can also be covered with the model from (5). In this case, $\mathbf{z}_i \in \mathbb{R}^{n_i}$ and the size of the network's state vector is $Z = \sum_{i=1}^{N} n_i$. In addition, for every pair of nodes v_i, v_j, the weight $\mathbf{w}_{ij} \in \mathbb{R}^{n_i \times n_j}$, while sparsity pattern Ψ describes the constraints on the weight matrices (for example, $\mathbf{w}_{ij} = \mathbf{0} \in \mathbb{R}^{n_i \times n_j}$ if $v_j \notin \mathcal{N}_{v_i} \cup \{v_i\}$).

A. Mapping Discrete-Time Controllers into the WCN

The fact that the WCN acts as a structured dynamical controller allows for the use of the network as computational resource. Compositionality of the WCN makes it suitable for control of large-scale systems, such as industrial process control and building automation systems. Thus, a procedure to map existing controllers into this computational substrate would enable a direct utilization of the well-known controllers already deployed in practice. In addition, it would allow for the use of WCNs in hierarchical control systems, where upper levels of control implement optimization based procedures (e.g., Model Predictive Control) to determine set points for the low level controllers that, in our case, would be implemented using the WCN. This way, the implementation and compositional benefits of the WCN would be married with the performance benefits of pre-designed centralized controllers (e.g., the use of the existing controller tuning algorithms for WCN configurations).

Although the WCN acts as a linear dynamical controller (5), the challenge is that the sparsity constraints imposed by the network topology prevent the controller from being mapped directly to the WCN. However, this constraint is counteracted by the fact that the state vector of the WCN is of the size Z, which may be significantly larger than the size of the state vector for the original controller. We will leverage these additional degrees of freedom to perform the mapping (see Section V). As shown in the next section, to find a configuration (i.e., matrices $\mathbf{W}, \mathbf{G}$ and $\mathbf{H}$) that causes

---

[1]In this work, the link weights w_{ij}, h_{ij} and g_{ij} for all nodes in the network are referred to as a *WCN configuration*.

the WCN to mimic the behavior of a given controller, we will first define appropriate metrics to measure 'closeness'. In this work, our goal is to minimize the widely applied $\mathbf{H}_\infty$ metric, and map the problem into optimization of structured linear controllers [12].

III. STRUCTURED LINEAR CONTROLLER REALIZATION

In this section we present an approach to formulate our design problem as an optimization problem. We focus on control of discrete-time LTI plants of the form:

$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{B}\mathbf{u}[k] + \mathbf{B}_d\mathbf{u}_d[k]$$
$$\mathbf{y}[k] = \mathbf{C}\mathbf{x}[k], \tag{6}$$

where $\mathbf{u}[k] \in \mathbb{R}^m$ denotes plant inputs (provided by actuators $a_1, ..., a_m$), $\mathbf{u}_d[k] \in \mathbb{R}^m$ specifies disturbance inputs, $\mathbf{y}[k] \in \mathbb{R}^p$ denotes measurements of the plant state vector $\mathbf{x} \in \mathbb{R}^n$ (provided by the sensors $s_1, ..., s_p$), while matrices $\mathbf{A}, \mathbf{B}, \mathbf{B_d}, \mathbf{C}$ have appropriate dimensions.

In this work, we investigate the problem of approximating any discrete LTI controller $\Sigma_1 = (\mathbf{A_1}, \mathbf{B_1}, \mathbf{C_1}, \mathbf{D_1})$, with a structured linear controller $\Sigma_2 = (\mathbf{A_2}, \mathbf{B_2}, \mathbf{C_2}, \mathbf{D_2})$ that may have a higher order. Note that for the structured linear controller we assume that the new controller has to satisfy structural constraints Ψ_2 imposed by the computational substrate. With Σ_c^1, Σ_c^2 we specify the closed-loop systems, where the plant is controlled by the controller Σ_1 and Σ_2, respectively. By denoting its state as $\mathbf{x}_c^i[k] = \begin{bmatrix} \mathbf{x}[k]^T & \mathbf{x}_i[k]^T \end{bmatrix}^T$, the closed loop system Σ_c^i ($i = 1, 2$) can be described as:

$$\mathbf{x}_c^i[k+1] = \begin{bmatrix} \mathbf{A} + \mathbf{B}\mathbf{D}_i\mathbf{C} & \mathbf{B}\mathbf{C}_i \\ \mathbf{B}_i\mathbf{C} & \mathbf{A}_i \end{bmatrix} \mathbf{x}_c^i[k] + \begin{bmatrix} \mathbf{B_d} \\ \mathbf{0} \end{bmatrix} \mathbf{u_d}[k]$$
$$\triangleq \mathbf{A}_c^i\mathbf{x}_c^i[k] + \mathbf{B}_c^i\mathbf{u_d}[k],$$
$$\mathbf{y}_c^i[k] = \begin{bmatrix} \mathbf{C} & \mathbf{0} \end{bmatrix} \mathbf{x}_c^i[k] \triangleq \mathbf{C}_c^i\mathbf{x}_c^i[k]$$

Now, we can formulate the problem as synthesis of a controller $(\mathbf{A_2}, \mathbf{B_2}, \mathbf{C_2}, \mathbf{D_2}) \in \Psi_2$ that minimizes the difference between the performances of the closed-loop systems Σ_c^1 and Σ_c^2 (i.e., when the plant is controlled by the initial or the structured controller). To evaluate the error or distance between the two closed-loop systems we employ the standard $\mathbf{H}_\infty$-norm, which is widely used in model reduction. This allows us to formulate the mapping problem as minimization of the approximation error between two systems. Exploiting the similarities between our problem and the model reduction problem, we start by using the similar approach as in [9] to construct the error system $\mathbf{\Sigma}_{ce}$. This system is described by the state $\mathbf{x}_{ce} = \begin{bmatrix} \mathbf{x}_c^{1T} & \mathbf{x}_c^{2T} \end{bmatrix}^T$ and transfer function $\mathbf{H}_{ce}(z) = \mathbf{H}_c^1(z) - \mathbf{H}_c^2(z)$, where $\mathbf{H}_c^1(z)$ and $\mathbf{H}_c^2(z)$ are the transfer functions of Σ_c^1 and Σ_c^2, respectively. The error system $\mathbf{\Sigma}_{ce}$ evolves as:

$$\mathbf{x}_{ce}[k+1] = \mathbf{A}_{ce}\mathbf{x}_{ce}[k] + \mathbf{B}_{ce}\mathbf{u_d}[k]$$
$$\mathbf{y}_{ce}[k] = \mathbf{C}_{ce}\mathbf{x}_{ce}[k], \tag{7}$$

where

$$\mathbf{A}_{ce} = \begin{bmatrix} \mathbf{A}_c^1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_c^2 \end{bmatrix}, \mathbf{B}_{ce} = \begin{bmatrix} \mathbf{B}_c^1 \\ \mathbf{B}_c^2 \end{bmatrix}, \mathbf{C}_{ce} = \begin{bmatrix} \mathbf{C}_c^1 & -\mathbf{C}_c^2 \end{bmatrix}. \tag{8}$$

The $\mathbf{H}_\infty$ norm of a stable discrete time system is defined as $\|\mathbf{\Sigma_{ce}}\|_{\mathcal{H}_\infty} = \sup \bar{\sigma}(\mathbf{H_{ce}}(e^{j\omega}))$, where $\bar{\sigma}$ is the maximum singular value of the matrix $\mathbf{H_{ce}}(e^{j\omega})$. Note that the error system parameters described as $\mathbf{\Sigma}_{ce} = (\mathbf{A}_{ce}, \mathbf{B}_{ce}, \mathbf{C}_{ce}, \mathbf{0})$ present a linear mapping of the structured controller parameters $\mathbf{\Sigma}_2 = (\mathbf{A}_2, \mathbf{B}_2, \mathbf{C}_2, \mathbf{D}_2)$.

Consequently, the problem of deriving feasible controller variables $(\mathbf{A_2}, \mathbf{B_2}, \mathbf{C_2}, \mathbf{D_2}) \in \Psi_2$, while minimizing the $\mathbf{H}_\infty$ norm of the error system can be specified as the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \gamma \\ \text{subject to} \quad & \|\mathbf{\Sigma_{ce}}\|_{\mathcal{H}_\infty} \leqslant \gamma, \\ & (\mathbf{A_2}, \mathbf{B_2}, \mathbf{C_2}, \mathbf{D_2}) \in \Psi_2 \end{aligned} \tag{9}$$

IV. APPROXIMATION ALGORITHMS

Since we have been able to formulate the problem of mapping discrete-time controllers into structured substrates as an optimization problem, in this section we continue by transforming the constraint for H_∞ norm of the error system to an LMI. We start by introducing the following result:

Lemma 1 ([17]): For asymptotically stable system (8), the following statements are equivalent:

(i) There exist matrices $\mathcal{X}, \Upsilon$ and $(\mathbf{A_2}, \mathbf{B_2}, \mathbf{C_2}, \mathbf{D_2}) \in \Psi_2$ such that

$$\begin{aligned} \mathcal{X} &\succ \mathbf{A}_{ce}\mathcal{X}\mathbf{A}_{ce}^T + \mathbf{B}_{ce}\mathbf{B}_{ce}^T \\ \Upsilon &\succ \mathbf{C}_{ce}\mathcal{X}\mathbf{C}_{ce}^T \end{aligned} \tag{10}$$

(ii) There exist matrices $\mathcal{X}, \Upsilon, \mathcal{Z}$ and $(\mathbf{A_2}, \mathbf{B_2}, \mathbf{C_2}, \mathbf{D_2}) \in \Psi_2$ such that

$$\begin{bmatrix} \mathcal{X} & \mathcal{Z} \\ \mathcal{Z} & \Upsilon \end{bmatrix} \succ \begin{bmatrix} \mathbf{A}_{ce} & \mathbf{B}_{ce} \\ \mathbf{C}_{ce} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathcal{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A}_{ce} & \mathbf{B}_{ce} \\ \mathbf{C}_{ce} & \mathbf{0} \end{bmatrix}^T \tag{11}$$

□

From Lemma 1, we construct the bound of the 'performance' of the error system.

Theorem 1 ([17]): Suppose that $\Upsilon \preceq \gamma\mathbf{I}$, where $\gamma \geq 0$, then $\|\mathbf{\Sigma_{ce}}\|_{\mathcal{H}_\infty} \leqslant \sqrt{\gamma}$ if and only if there exist $\mathcal{X} \succ 0$, $\Upsilon \succ 0$, $\mathcal{Z} = 0$ such that (11) holds, which is equivalent to:

$$\mathcal{R}(\mathcal{X}, \mathbf{0}, \Upsilon, \mathcal{X}^{-1}) = \begin{bmatrix} \mathcal{X} & \mathbf{0} & \mathbf{A_{ce}} & \mathbf{B_{ce}} \\ \mathbf{0} & \Upsilon & \mathbf{C_{ce}} & \mathbf{0} \\ \mathbf{A_{ce}^T} & \mathbf{C_{ce}^T} & \mathcal{X}^{-1} & \mathbf{0} \\ \mathbf{B_{ce}^T} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \succ 0 \tag{12}$$

□

Therefore, our objective is to derive $(\mathbf{A_2}, \mathbf{B_2}, \mathbf{C_2}, \mathbf{D_2}) \in \Psi_2$, along with feasible $\mathcal{X}, \Upsilon$ that minimize the scalar variable γ under the constraints specified in Theorem 1.

A. Linearization Algorithm

From Theorem 1, the obtained optimization problem has only one non-convex term ($\mathcal{X}^{-1}$), which is used to specify the constraint in (12). To be able to solve the problem, we approximate the term with a suitable convex expression. One convexifying, and more specifically, a linearization method 'around' any matrix $\mathcal{X}_k$ (described in [14], [12]) is:

$$LIN(\mathbf{X}^{-1}, \mathbf{X_k}) = \mathbf{X_k}^{-1} - \mathbf{X_k}^{-1}(\mathbf{X} - \mathbf{X_k})\mathbf{X_k}^{-1}. \tag{13}$$

This linearization provides a sufficient (and conservative) condition for the optimal solution, while enabling us to specify the constraint (12) as an LMI. Furthermore, we can now extend the approach from [12] to define iterative algorithms for embedding linear controllers into structured substrates.

B. Initialization and Optimal Algorithms

According to (8), the set of eigenvalues of $\mathbf{A}_{ce}$ is the union of eigenvalues of $\mathbf{A}_c^1$ and $\mathbf{A}_c^2$. Thus, Σ_{ce} is stable if and only both $\mathbf{A}_c^1$ and $\mathbf{A}_c^2$ are stable. We assume that the initial controller guarantees stability of the closed-loop system Σ_c^1, and that there exists a stable configuration for the imposed structural constraints.[2] Therefore, there exist feasible $\mathcal{X}, \Upsilon$, and $(\mathbf{A_2}, \mathbf{B_2}, \mathbf{C_2}, \mathbf{D_2}) \in \Psi_2$, such that $\sqrt{\gamma} = \|\mathbf{\Sigma_c}\|_{\mathcal{H}_\infty}$ is the optimal solution (i.e., cost) of the problem (9).

To solve the optimization problem we define the following iterative algorithms (Algorithm 1 and 2). Since the linearization $LIN(\mathbf{X}^{-1}, \mathbf{X_k})$ provides a sufficient problem constraint, the iterative optimization procedure specified in Algorithm 1 may not work with a randomly generated initial point $\mathbf{X_k}$ (i.e., for $k = 0$). Therefore, to obtain a feasible initial point for Algorithm 1, we replace $\mathbf{X}^{-1}$ in (12) with a new variable ($\mathbf{Y}$), and relax the constraint first. This relaxation results in the initialization Algorithm 2 that starts from a random selected matrix $\mathbf{X}_0$ (i.e., $\mathbf{X}_0 = \mathbf{I} + \mathbf{R}^T\mathbf{R}$, where $\mathbf{R}$ is a random matrix with appropriate dimensions).

Algorithm 1 Mapping linear controllers into structured substrate

1) Set $\epsilon > 0$, $k_{stop} > 0$, and $k = 0$.
2) Solve the following convex optimization problem.

  minimize γ

  subject to $\mathcal{R}(\mathcal{X}, \mathbf{0}, \Upsilon, LIN(\mathcal{X}^{-1}, \mathcal{X}_k)) \succ 0,$

$\qquad\qquad \mathcal{X} \succ 0, \Upsilon \succ 0, \Upsilon \preceq \gamma\mathbf{I}, (\mathbf{A_2}, \mathbf{B_2}, \mathbf{C_2}, \mathbf{D_2}) \in \Psi_2,$

3) If $\gamma < \epsilon$ or $k > k_{stop}$, stop. Otherwise, set $k = k + 1$, $\mathcal{X}_{k+1} = \mathcal{X}$ and go back to step 2.

The convexifying and linearization method ensures that at each iteration k, the optimal solution γ_{k+1} satisfies that $0 \leq \gamma_{k+1} \leq \gamma_k$, so the sequence converges to a local optimal solution [12].

It is worth noting here that the problem formulation and algorithms derived in this paper have been based on the closed-loop system model. Another possible approach would be to reduce the distance between the two controller systems directly. However, this would require that the initial (i.e., given) controller is stable.

V. APPROXIMATING PID WITH THE WCN

PID controllers are the most commonly used type of linear controllers. Therefore, in this section, we illustrate how to

---

Algorithm 2 Iterative algorithm used to obtain a feasible initial point

1) Set $\epsilon > 0$, and $k = 0$.
2) Solve the following convex optimization problem.

  minimize γ

  subject to $\mathcal{R}(\mathcal{X}, \mathbf{0}, \Upsilon, \mathcal{Y}) \succ 0,$

$\qquad\qquad \begin{bmatrix} \mathcal{X} & \mathbf{I} \\ \mathbf{I} & \mathcal{Y} \end{bmatrix} \succeq 0, \Upsilon \preceq \gamma\mathbf{I}, \mathcal{X} \succ 0, \mathcal{Y} \succ 0, \Upsilon \succ 0,$

$\qquad\qquad \mathcal{Y} \preceq LIN(\mathcal{X}^{-1}, \mathcal{X}_k) + \gamma\mathbf{I},$

$\qquad\qquad (\mathbf{A_2}, \mathbf{B_2}, \mathbf{C_2}, \mathbf{D_2}) \in \Psi_2,$

3) If $\gamma < \epsilon$, $\|\lambda_{\mathbf{A}_{ce}}\| < 1$,[3]stop. Otherwise, set $k = k + 1$, $\mathcal{X}_{k+1} = \mathcal{X}$ and go back to step 2.
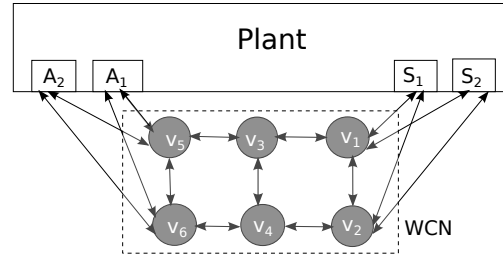

Fig. 2: An example of the WCN topology with 6 nodes, where each node can maintain a scalar or vector state.

employ the algorithms from the previous section to synthesize WCN configurations. The goal is to minimize the error between initial Multiple-Input Multiple-Output (MIMO) PID controllers and the derived WCNs. We consider the case when the initial linear PID controller has states $\mathbf{x}_1 \in \mathbb{R}^4$, while the WCN scheme is implemented on the network with the topology shown in Figure 2. Furthermore, the controlled plant has states $\mathbf{x} \in \mathbb{R}^4$, two inputs and two outputs.

Initially, we consider the *scalar WCN* setup where each node in the WCN maintains a scalar state, i.e., $z_i[k] \in \mathbb{R}$ in (3). Consequently, in this case the WCN acts as a structured dynamical controller with $\mathbf{z}[k] \in \mathbb{R}^6$ in (5). Using Algorithms 1 and 2, which were implemented using CVX, a package for specifying and solving convex optimization programs [18], after less than 60 iterations we obtained $\gamma = 0.1294$, meaning that $\|\mathbf{\Sigma_{ce}}\|_{\mathcal{H}_\infty} = \sqrt{\gamma} \approx 0.3597$ (see Figure 3). In addition, we compared the step (Figure 4(a)) and frequency (Figure 4(b)) responses for the closed-loop systems controlled by the initial PID and the obtained WCN configuration.

From the above results, we can conclude that the structural constraints for a scalar state WCN (from Figure 2) imposed a significant limitation on our capabilities to approximate behavior of the initial MIMO PID controller. However, the error between these closed-loop systems (i.e., Σ_c^1 and Σ_c^2) can be decreased if each node in the WCN maintains a larger state (i.e., a vector state). We will refer to these systems as *vector WCNs*. By incorporating different memory and computing capabilities to each node, the vector WCN where each node v_i maintains a (vector) state from $\mathbb{R}^{n_i}$ can be modeled as in (3), (4), where $\mathbf{z_i} \in \mathbb{R}^{n_i}, n_i > 1$, $\mathbf{w_{ij}} \in$
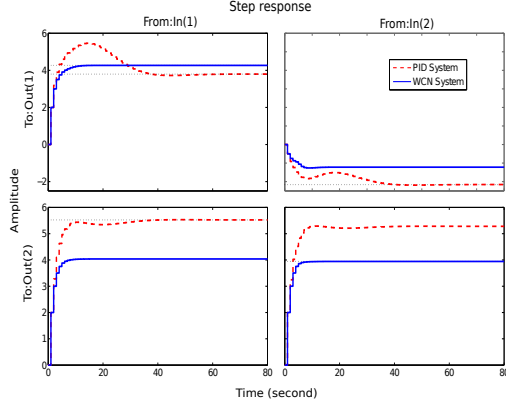
(a) Step response
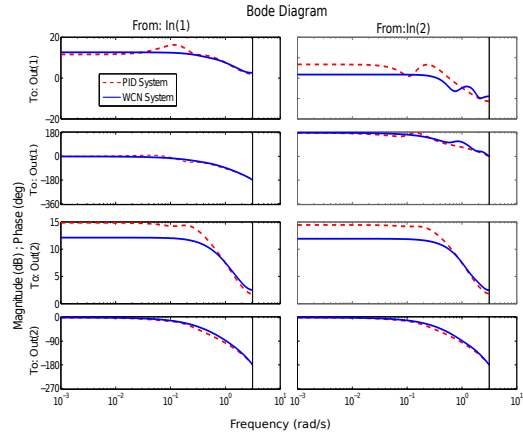
(b) Frequency response

Fig. 4: Comparison of the closed-loop systems controlled by the initial PID and the WCN, where each node in the network from Figure 2 maintains a scalar state. Final system approximation error is $\gamma = 0.1294$, $\|\mathbf{\Sigma_{ce}}\|_{\mathcal{H}_\infty} = \sqrt{\gamma} \approx 0.3597$.
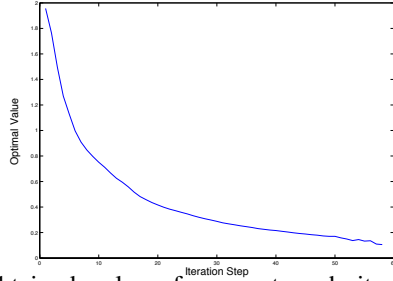

Fig. 3: Obtained values for γ_k at each iteration step of Algorithm 1, when each node in the WCN from Figure 2 maintains a scalar state.

$\mathbb{R}^{n_i \times n_j}$, $\mathbf{h_{ij}} \in \mathbb{R}^{n_i}$, $\mathbf{g_{ij}} \in \mathbb{R}^{1 \times n_j}$.

Algorithms 1 and 2 can be easily extended to cover this vector state scenario, as in this case only the structural constraints have to be generalized. A straightforward extension would require running both Algorithms 2 and 1 to obtain initial (feasible) and optimal points. However, it is possible to significantly speed up the initialization procedure for vector WCNs. This is achieved by running Algorithm 2 for the scalar WCN with the same topology, and then expanding the obtained feasible configuration to derive an initialization point for the vector WCN. More precisely, assume Algorithm 2 provides a scalar WCN configuration $(\mathbf{W^0}, \mathbf{H^0}, \mathbf{G^0})$ with closed-loop transfer function H. Then, an initial feasible vector WCN configuration with closed-loop transfer function H, and where nodes maintain states of size $\tilde{n}$, can be derived from the vector weights defined using the initial scalar weights as:

$$\mathbf{w}_{ij}^1 = w_{ij}^0 \mathbf{I}_{\tilde{n}}, \mathbf{h}_{ij}^1 = c_{\tilde{n}}(h_{ij}^0), \mathbf{g}_{ij}^1 = \frac{1}{\tilde{n}} r_{\tilde{n}}(g_{ij}^0). \qquad (14)$$

Here, $c_{\tilde{n}}(a)$ $(r_{\tilde{n}}(b))$ denotes a size $\tilde{n}$ column (row) vector where every element equals to a (b), while $\mathbf{I}_{\tilde{n}}$ is the identity matrix of size $\tilde{n}$. From (14), a new feasible error system $\Sigma_{ne} = (\mathbf{A}_{ne}, \mathbf{B}_{ne}, \mathbf{C}_{ne})$ (from (7), (8)) can be obtained.

Therefore, by solving a **single** convex optimization problem (15), instead of a sequence of optimization problems (Algorithm 2), we can obtain a feasible initial point $\mathcal{X}_1$ for

the Algorithm 1 of vector WCN

$$
\begin{aligned}
\text{minimize} \quad & \gamma_0 \\
\text{subject to} \quad & \mathcal{X}_1 \succeq \mathbf{A}_{ne}\mathcal{X}_1\mathbf{A}_{ne}^T + \mathbf{B}_{ne}\mathbf{B}_{ne}^T, \\
& \Upsilon \succeq \mathbf{C}_{ne}\mathcal{X}_1\mathbf{C}_{ne}^T, \\
& \mathcal{X}_1 \succ 0, \Upsilon \preceq \gamma_0 \mathbf{I}.
\end{aligned}
\qquad (15)
$$

To summarize, the speed up is obtained by avoiding to directly use Algorithm 2 to derive a feasible vector WCN configuration, as this would require solving a large convex optimization problem at each iteration. Instead, we iteratively solve a smaller problem (for scalar WCN configurations).

Note that the result γ_0 of the optimization problem (15) is an upper bound for the approximation error when a vector WCN is used, since the Algorithm 1 provides a decreasing sequence of γ_k's after the initialization step. Table 1 presents a comparison of the times required to compute the initial points for Algorithm 1 when each node maintains a state $\mathbf{z}_i \in \mathbb{R}^3$ in the WCN from Figure 2. We compared the following methods: 1) when the Algorithm 2 is used to obtain a scalar WCN configuration, then extended into the 'vector' WCN by (14), (15); or 2) when Algorithm 2 is used to directly compute a feasible vector configuration. As can be seen, both the time of every iteration step and the number of iteration steps are significantly increased when the 'vector' (i.e., direct) form of Algorithm 2 is used. This is caused by the fact that the size of the problem directly affects the performance of the LMI solver (i.e., CVX).

Finally, we showed that this more powerful, vector WCN, can reduce the approximation error. Figure 5(a) and Figure 5(b) show comparisons of the closed-loop system performances when the plant is controlled by the initial PID controller and the WCN where each node maintains a state $\mathbf{z}_i \in \mathbb{R}^3$. In this case, using the aforementioned procedure we obtained $\gamma = 0.0823$, thus reducing the approximation error by almost 50%.

VI. CONCLUSION

In this paper, we have presented a method for mapping discrete-time linear controllers into structured computation
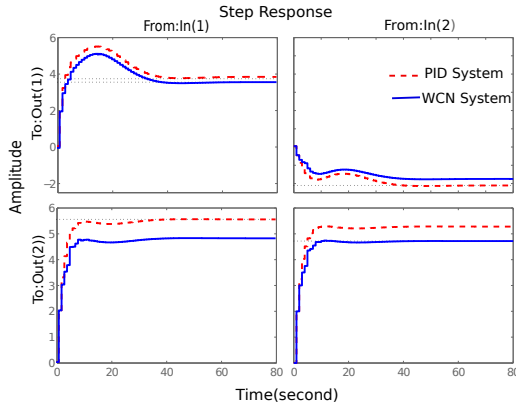
(a) Step response
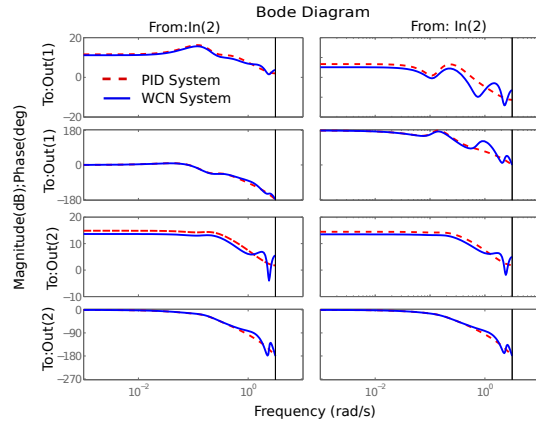
(b) Frequency response

Fig. 5: Comparison of the closed-loop systems controlled by the initial PID and the WCN, where each node in the network from Figure 2 maintains a state from $\mathbb{R}^3$. Algorithm 1 gets $\gamma = 0.0823$, and the approximation error between two closed loop systems are smaller than what we get from the scalar state WCN (each node maintains a state from $\mathbb{R}$).

Method to derive the initial vector WCN configuration	Number of iteration steps	Time for each step
Expand a feasible 'scalar' configuration	58	1s
Direct use of Algorithm 2	675	9s

TABLE I: comparison of computation speed

substrates. This problem has been motivated by the Wireless Control Network, a recently introduced distributed scheme for control over multi-hop wireless networks. By exploiting the similarities with the model reduction problem, we have formulated the structured approximation problem as an optimal design of structured linear controllers, and specified the algorithms that can be used for the networked system realization. In addition, we have illustrated their use on the mapping of MIMO PID controllers into the WCN, and showed how to reduce the computation overhead of the mapping when the WCN nodes maintain vector states. Finally, we have shown that an increase in sizes of the states maintained by the nodes in the WCN, decreases the approximation error. However, it would be beneficial to provide a way to estimate mapping improvements with the increase of the nodes' state sizes. This is an avenue for future work.

REFERENCES

[1] M. Pajic, S. Sundaram, G. Pappas, and R. Mangharam, "The wireless control network: A new approach for control over networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2305 –2318, 2011.

[2] O. C. Imer, S. Yuksel, and T. Basar, "Optimal control of LTI systems over unreliable communication links," *Automatica*, vol. 42, no. 9, pp. 1429–1439, 2006.

[3] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, "Foundations of control and estimation over lossy networks," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 163–187, 2007.

[4] V. Gupta, A. F. Dana, J. Hespanha, R. M. Murray, and B. Hassibi, "Data transmission over networks for estimation and control," *IEEE Transactions on Automatic Control*, vol. 54, no. 8, pp. 1807–1819, 2009.

[5] B. Bamieh, F. Paganini, and M. Dahleh, "Distributed control of spatially invariant systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 7, pp. 1091 –1107, 2002.

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

[7] M. Rotkowitz and S. Lall, "A characterization of convex problems in decentralized control," *IEEE Transactions on Automatic Control*, vol. 51, no. 2, pp. 274 –286, 2006.

[8] A. Vamsi and N. Elia, "Optimal realizable networked controllers for networked systems," in *American Control Conference (ACC), 2011*, 2011, pp. 336–341.

[9] P. Van Dooren, K. A. Gallivan, and P.-A. Absil, "H_2-optimal model reduction of MIMO systems," *Applied Mathematics Letters*, vol. 21, no. 12, pp. 1267–1273, 2008.

[10] S. Gugercin and A. C. Antoulas, "A survey of model reduction by balanced truncation and some new results," *International Journal of Control*, vol. 77, no. 8, pp. 748–766, 2004.

[11] K. Grigoriadis, "Optimal H_∞ model reduction via linear matrix inequalities: continuous and discrete-time cases," in *Proceedings of the 34th IEEE Conference on Decision and Control*, vol. 3, 1995, pp. 3074 –3079.

[12] J. Han and R. Skelton, "An LMI optimization approach for structured linear controllers," in *Proc. 42nd IEEE Conference on Decision and Control*, vol. 5, 2003, pp. 5143 – 5148.

[13] L. El Ghaoui, F. Oustry, and M. AitRami, "A cone complementarity linearization algorithm for static output-feedback and related problems," *IEEE Transactions on Automatic Control*, vol. 42, no. 8, pp. 1171 –1176, 1997.

[14] M. de Oliveira, J. Camino, and R. Skelton, "A convexifying algorithm for the design of structured linear controllers," in *Proceedings of the 39th IEEE Conference on Decision and Control*, vol. 3, 2000, pp. 2781 –2786.

[15] 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 *Proceedings of the 11th international conference on Information Processing in Sensor Networks*, ser. IPSN '12, 2012, pp. 25–36.

[16] M. Pajic, S. Sundaram, G. J. Pappas, and R. Mangharam, "Topological conditions for in-network stabilization of dynamical systems," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 4, pp. 1–15, 2013.

[17] R. E. Skelton, T. Iwasaki, and K. Grigoriadis, *A unified algebraic approach to linear control design*. CRC Press, 1998.

[18] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, ver. 2.0," http://cvxr.com/cvx, 2012.