Composability and Controllability of Structural Linear Time-Invariant Systems: Distributed Verification ⋆

J. Frederico Carvalho [f,a], Sérgio Pequito [b,c,a], A. Pedro Aguiar [d], Soummya Kar [b], Karl H. Johansson [e]

[a] *Both authors contributed equally to this work*

[b] *Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213*

[c] *Institute for Systems and Robotics, Instituto Superior Técnico, Technical University of Lisbon, Lisbon, Portugal*

[d] *Department of Electrical and Computer Engineering, Faculty of Engineering, University of Porto, Porto, Portugal*

[e] *ACCESS Linnaeus Center, School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden*

[f] *Center for Autonomous Systems CAS/RPL , KTH Royal Institute of Technology, Stockholm, Sweden*

Abstract

Motivated by the development and deployment of large-scale dynamical systems, often comprised of geographically distributed smaller subsystems, we address the problem of verifying their controllability in a distributed manner. Specifically, we study controllability in the structural system theoretic sense, *structural* controllability, in which rather than focusing on a specific numerical system realization, we provide guarantees for equivalence classes of linear time-invariant systems on the basis of their structural sparsity patterns, i.e., the location of zero/nonzero entries in the plant matrices. Towards this goal, we fist provide several necessary and/or sufficient conditions that ensure that the overall system is structurally controllable on the basis of the subsystems' structural pattern and their interconnections. The proposed verification criteria are shown to be efficiently implementable (i.e., with polynomial time-complexity in the number of the state variables and inputs) in two important subclasses of interconnected dynamical systems: *similar* (where every subsystem has the same structure) and *serial* (where every subsystem outputs to at most one other subsystem). Secondly, we provide an iterative distributed algorithm to verify structural controllability for general interconnected dynamical system, i.e., it is based on communication among (physically) interconnected subsystems, and requires only local model and interconnection knowledge at each subsystem.

Key words: Control system analysis, Controllability, Structural properties, Graph theory, Combinatorial mathematics

1 Introduction

In recent years we have witnessed an explosion in the use of large-scale dynamical systems, notably, those with a modular structure (Özgüner and Hemani, 1985; Davison and Özgüner, 1983; Davison, 1977), such as content delivery networks, social networks, robot swarms, and smart grids. Such systems, often geographically distributed, are comprised of smaller subsystems (which we may refer to as *agents*), and a typical concern is ensuring that the system, as a whole, performs as intended. More than often, when analyzing these interconnected dynamical systems, which in this paper we consider to consist of continuous linear-time invariant (LTI) subsystems, we do not know the exact parameters of the plant matrices. Therefore, we focus on the zero/nonzero pattern of the system's plant, which we refer to as *sparsity pattern*, and we focus on structural counterpart of controllability, i.e., *structural controllability* (Dion et al., 2003).

It is worthwhile noting that these agents may be *homogeneous* or *heterogeneous*, from its structure point of view.

⋆ This work was partially supported by grant SFRH/BD/33779/2009, from Fundação para a Ciência e a Tecnologia (FCT) and the CMU-Portugal (ICTI) program, and by projects POCI-01-0145-FEDER-006933/SYSTEC funded by FEDER funds through COMPETE2020 and FCT. The work by K. H. Johansson was supported by the Knut and Alice Wallenberg Foundation, the Swedish Strategic Research Foundation, and the Swedish Research Council.

Email addresses: jfpbdc@kth.se (J. Frederico Carvalho), sergio.pequito@gmail.com (Sérgio Pequito), pedro.aguiar@fe.up.pt (A. Pedro Aguiar), soummyak@andrew.cmu.edu (Soummya Kar), kallej@kth.se (Karl H. Johansson).

When the agents are homogeneous, their plants and connections (when used) have the same sparsity pattern and the system is referred to as a *similar system*. Otherwise, the agents are heterogeneous and two possible scenarios are conceivable: (*i*) an agent may receive information from (possibly several) other agents but it only transmits to one other agent, the overall system is referred to as *serial*, and commonly arises in peer-to-peer communication schemes; and (*ii*) the communications between agents can be arbitrary, which commonly arise in broadcast communication setups. All the above subclasses of interconnected dynamical systems are of interest and explored in detail in this paper. More precisely, we provide several necessary and/or sufficient conditions to ensure key properties of the system, which can be verified resorting to efficient (i.e., with polynomial time complexity in the number of state variables) algorithms.

In some applications, the problem of composability is particularly relevant. Consider, for example, a swarm of robots possessing similar structure where the communication topology may change over time, or where robots may join or leave the swarm over time. Then, the existence of necessary and/or sufficient conditions on the structure and interconnection between these agents contribute to *controllability-by-design* schemes, i.e., we ensure that by inserting an agent into the interconnected dynamical system, we obtain a controllable dynamical system. Consequently, we can specify with which agents an agent should interact with such that those conditions hold.

A swarm of robots can also be composed by a variety of heterogeneous agents in which case controllability-by-design is also important, yet due to constraints on the communication range, the interaction between agents is merely local, even if some additional information is known. Therefore, in the context of serial systems we can equip each subsystem with the capability of inferring if the entire system is structurally controllable, i.e., we provide distributed algorithms that rely only on the interaction between a subsystem and its neighbors, where information about their structure may be shared. In particular, if we equip the robots in the swarm with actuation capabilities that can be activated when the interconnected dynamical system is not structurally controllable, we can render this interconnected dynamical system structurally controllable.

Nonetheless, imposing *a priori* knowledge of the structure of the interconnections in the system (for instance, wether it is a serial system) can be restrictive, so distributed algorithms to verify structural controllability of general interconnected dynamical systems are in need. Hereafter, we provide such an algorithm: It requires the interaction between a subsystem and its neighbors, but it does not require to share the structure of the subsystems involved. Instead, it requires only partial information about its structure, which leads to a certain level of privacy of the intervenients in the communication. The proposed scheme is also particularly suitable to other applications such as the smart grid of the future, that consists of entities described by subsystems deployed over large distances; in particular, notice that in these cases, the different entities may not be willing to share information about their structure due to security or privacy reasons.

Related Work: Structural controllability was introduced by Lin (1974) in the context of single-input single-output (SISO) systems, and extended to multi-input multi-output (MIMO) systems by Shields and Pearson (1976). A recent survey of the results in structural systems theory, where several necessary and sufficient conditions are presented, can be found in Dion et al. (2003).

In this paper, we focus on the composability aspects that ensure structural controllability. In other words, we are interested in understanding how the connection between different dynamical subsystems enables or jeopardizes the structural controllability of the overall system. The presented problem statement fits the general framework presented in Anderson and Hong (1982). Nevertheless, the verification procedures proposed in Anderson and Hong (1982) based in matrix nets lead to a computational burden which increases exponentially with the dimension of the problem. Alternatively, in Davison (1977) an efficient method is proposed that takes into account the whole system instead of local properties (i.e., the components of the system and their interconnections), however this method does not apply to an arbitrary systems. More precisely, it is assumed that when connected, the state space digraph (to be defined later) is spanned by a disjoint union of cycles, which is called a *rank constraint*. In contrast, in Rech and Perret (1991) and Li et al. (1996), the authors have presented results on the structural controllability of interconnected dynamical systems, by focusing on the cascade interconnection of system structures that ensure the structural controllability of the interconnected dynamical system. Nevertheless, these structures are not unique, and the interconnection of these is established assuming such connectible structures are given, therefore, no practical criteria to compute the structures and verify the results is given. More recently, in Blackhall and Hill (2010) similar results were obtained by exploring which variables may belong to a structure and referred to as controllable state variable. Thus, similarly to Rech and Perret (1991) and Li et al. (1996), the results depend on the identified structures, but no method to systematically identify these structures is provided. In Yang and Zhang (1995) the study is conducted assuming that all the subsystems except a *central* subsystem, which is allowed to communicate with every other subsystem, have the same dynamic structure, and the interconnection between the several subsystems also has the same structure (even though they may not be used). This, study considers the impact of local interactions into the system structural controllability, which results can be obtained with the solution proposed hereafter.

In Pequito et al. (2016*a*), we studied the problem of determining the sparsest input matrix to ensure structural controllability in a centralized fashion. Furthermore, polynomial algorithms with computational complexity $\mathcal{O}(n^3)$ were provided to both problems, where n is the number of state variables. In Pequito et al. (2015), we studied the setting where the selection of inputs is constrained to a given collection, and shown to be NP-hard. Finally, in Pequito et al. (2016*b*), the problem in Pequito et al. (2016*a*) was further extended to determining the input matrix incurring in the minimum cost when the state variables actuated incur in different costs while ensuring structural controllability. Furthermore, procedures with $\mathcal{O}(n^\omega)$ computational complexity were provided, where $\omega < 2.373$ is the lowest known exponent associated with the complexity of multiplying two $n \times n$ matrices. All these contrasts with the problem addressed in the current paper in the sense that we aim to verify structural controllability properties in a *distributed fashion*. In particular, it requires identifying specific network conditions on the network structure under which we can use efficient algorithms, i.e., polynomial in the dimension of the state space, or provide distributed algorithms suitable to address the proposed problem.

On the other hand, composability aspects regarding controllability have been heavily studied by several authors, see for instance, Zhou (2015); Chen and Desoer (1967); Wolovich and Hwang (1974); Yonemura and Ito (1972); Wang and Davison (1973); Davison and Wang (1975). Briefly, all these studies resort to the well known Popov-Belevotch-Hautus (PBH) eigenvalue controllability criterion for LTI systems (Hespanha, 2009). We notice that this criterion requires the knowledge of the overall system to infer its controllability. The reason is closely related with the loss of degrees of freedom imposed by interconnected dynamical systems, as well as conservation laws in general, that reflects in the decrease of the rank of the system's dynamics matrix when compared with the sum of the rank of the dynamics matrices of each subsystem. Consequently, even if all subsystems are controllable, their interconnection may not be. Notwithstanding, the same does not happen when dealing with structural systems, where if all subsystems are structurally controllable, then the overall system is structurally controllable. So, while not guaranteeing that a system is controllable, we can regard these as necessary conditions for controllability.

Main Contributions: The main contributions of this paper are threefold:

(i) we provide sufficient conditions for similar systems to be structurally controllable. More precisely, these rely only on the structure of the subsystem and interconnection between subsystems. A distributed algorithm is proposed, that can verify these conditions in polynomial time;

(ii) we provide sufficient conditions for serial systems to be structurally controllable. A distributed algorithm to verify these conditions is provided. It requires only the knowledge of the subsystem and its neighbors' structure, as well as its interconnections. This algorithm requires only the capability of a subsystem to communicate with its neighbors, and has computational complexity equal to $\mathcal{O}\left(n_S^3\right)$, where n_S corresponds to the total number of state variables and inputs present in a subsystem and its neighbors; and

(iii) we provide a distributed algorithm to verify necessary and sufficient conditions to ensure structural controllability for any interconnected dynamical system that consists of LTI subsystems. This algorithm requires only the capability of a subsystem to communicate with its neighbors, have access to its own structure and partial information regarding decisions performed by its neighbors that do not require sharing the structure of the neighboring agents.

The rest of this paper is organized as follows. In Section 2 we formally describe the problem statement. Section 3 introduces some concepts in structural systems theory, that will be used throughout the remainder of the paper. The main contributions are presented in Section 4, and in Section 5 we provide examples that illustrate the main findings. Finally, Section 6 concludes the paper and discusses avenues for further research.

2 Problem Statement

Consider r linear time-invariant (LTI) dynamical systems described by

$$\dot{x}_i(t) = A_i x_i(t) + B_i u_i(t), \quad i = 1, \ldots, r,$$

where $x_i \in \mathbb{R}^{n_i}$ is the state, and $u_i \in \mathbb{R}^{p_i}$ the input. The dynamical system can be described by the pair (A_i, B_i), where $A_i \in \mathbb{R}^{n_i \times n_i}$ is the dynamic matrix of subsystem i and $B_i \in \mathbb{R}^{n_i \times p_i}$ its input matrix. By considering the interconnection from subsystem i to subsystem j for all possible subsystems we obtain the interconnected dynamical system described as follows:

$$\dot{x}(t) = \underbrace{\begin{bmatrix} A_1 & E_{1,2} & \cdots & E_{1,r} \\ E_{2,1} & A_2 & \cdots & E_{2,r} \\ \vdots & \ddots & \ddots & \vdots \\ E_{r,1} & \cdots & E_{r,r-1} & A_r \end{bmatrix}}_{A} x(t) + \underbrace{\begin{bmatrix} B_1 & 0 & \cdots & 0 \\ 0 & B_2 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & B_r \end{bmatrix}}_{B} u(t) \qquad (1)$$

where the state is given by $x = [x_1^\intercal \ \ldots \ x_r^\intercal]^\intercal \in \mathbb{R}^n$, with $n = \sum_{i=1}^r n_i$, and the input given by $u = [u_1^\intercal \ \ldots \ u_r^\intercal]^\intercal \in$

$\mathbb{R}^p$, with $p = \sum_{i=1}^{r} p_i$. In addition, $E_{i,j} \in \mathbb{R}^{n_i \times n_j}$ is referred to as the *connection matrix* from the j–th subsystem to the i–th subsystem. Furthermore, we denote the system (1) by the matrix pair (A, B), denoting the i–th subsystem, $i = 1, \ldots, r$ of (1) by the matrix pair (A_i, B_i). Finally, we call those subsystems (A_j, B_j), with $j = 1, \ldots, r$ such that $E_{j,i} \neq 0$, the *outgoing neighbors* of the i–th subsystem, and those that $E_{i,j} \neq 0$ the *incoming neighbors* of the i–th subsystem; we refer to them collectively as the *neighbors* of the i–th subsystem.

Now, consider the sparsity pattern of matrix pair (A, B) which we denote by the structural system $(\bar{A}, \bar{B})$; similarly, we denote by $(\bar{A}_i, \bar{B}_i)$ the structural pair of matrices associated with (A_i, B_i), and $\bar{E}_{i,j}$ the sparsity pattern of $E_{j,i}$. Then, a structurally controllable system is defined as follows (Dion et al., 2003).

Definition 1 *Given a structural system $(\bar{A}, \bar{B})$, we say that it is structurally controllable if and only if, there exists at least one control system (A, B) with the same sparsity pattern as $(\bar{A}, \bar{B})$ (i.e. $A_{i,j} = 0$ if $\bar{A}_{i,j} = 0$ and $B_{i,k} = 0$ if $\bar{B}_{l,k} = 0$) which is controllable.* ⋄

It can be seen, from density arguments, that if $(\bar{A}, \bar{B})$ is structurally controllable, then almost all control systems (A, B) with the same sparsity as $(\bar{A}, \bar{B})$ are structurally controllable (Dion et al., 2003). We say that a control system (A, B) is *structurally controllable* if the associated structural system $(\bar{A}, \bar{B})$ is structurally controllable.

The problem addressed in the current paper can be posed as follows.

<u>*Problem:*</u> Given a collection of control systems (A_i, B_i), $i = 1, \ldots, r$, and the interconnection from the subsystem i to its neighbors, i.e., $(A_j, B_j, E_{j,i})$ for all $j \neq i$, design a distributed procedure to determine if the interconnected control system (A, B) given in (1) is structurally controllable. ∘

Furthermore, note that in a non-structural setting local properties are not enough to guarantee controllability, since the connection to other subsystems may lead to parameter cancellation (Wang and Davison, 1973; Davison and Wang, 1975); therefore, the approach presented hereafter allows us to obtain only necessary conditions for controllability.

3 Preliminaries and Terminology

In this section, we review some of the concepts used to analyze the problem of structural controllability of interconnected dynamical systems, which illustrations can be found in Pequito et al. (2016*a*).

In order to perform structural analysis efficiently, it is customary to associate to (1) a directed graph, or *digraph* $\mathcal{D} = (\mathcal{V}, \mathcal{E})$, in which $\mathcal{V}$ denotes the set of *vertices* and $\mathcal{E}$ the set of *edges*, where (v_j, v_i) represents an *edge from the vertex v_j to the vertex v_i*. To this end, let $\bar{A} \in \{0, 1\}^{n \times n}$ and $\bar{B} \in \{0, 1\}^{n \times p}$ be the binary matrices that represent the sparsity patterns of A and B as in (1), respectively. Denote by $\mathcal{X} = \{x_1, \cdots, x_n\}$ and $\mathcal{U} = \{u_1, \cdots, u_p\}$ the sets of *state* and *input vertices*, respectively, and by $\mathcal{E}_{\mathcal{X},\mathcal{X}} = \{(x_i, x_j) : \bar{A}_{ji} \neq 0\}$, $\mathcal{E}_{\mathcal{U},\mathcal{X}} = \{(u_j, x_i) : \bar{B}_{ij} \neq 0\}$, the sets of edges between the vertex sets in subscript. We may then introduce the *state digraph* $\mathcal{D}(\bar{A}) = (\mathcal{X}, \mathcal{E}_{\mathcal{X},\mathcal{X}})$ and the *system digraph* $\mathcal{D}(\bar{A}, \bar{B}) = (\mathcal{X} \cup \mathcal{U}, \mathcal{E}_{\mathcal{X},\mathcal{X}} \cup \mathcal{E}_{\mathcal{U},\mathcal{X}})$. Note that in the digraph $\mathcal{D}(\bar{A}, \bar{B})$, the input vertices representing the zero columns of $\bar{B}$ correspond to isolated vertices. As such, the number of *effective* inputs, i.e., the inputs which actually exert control, is equal to the number of nonzero columns of $\bar{B}$, or, in the digraph representation, the number of input vertices that are connected to at least one state vertex through an edge in $\mathcal{E}_{\mathcal{U},\mathcal{X}}$.

A *directed path* from the vertex v_1 to v_k is a sequence of edges $\{(v_1, v_2), (v_2, v_3), \ldots, (v_{k-1}, v_k)\}$. If all the vertices in a directed path are distinct, then the path is said to be an *elementary path*. A *cycle* is an elementary path from v_1 to v_k, together with an edge from v_k to v_1.

Given a digraph $\mathcal{D} = (\mathcal{V}, \mathcal{E})$, we say that $\mathcal{D}' = (\mathcal{V}', \mathcal{E}')$ is a subgraph of $\mathcal{D}$ if it is a digraph with $\mathcal{V}' \subseteq \mathcal{V}$ and $\mathcal{E}' \subseteq \mathcal{E}$, which we denote by $\mathcal{D}' \subseteq \mathcal{D}$. Furthermore, we say that $\mathcal{D}'$ spans $\mathcal{D}$ if $\mathcal{V}' = \mathcal{V}$.

We also require the following graph-theoretic notions (Cormen et al., 2001). A digraph $\mathcal{D}$ is strongly connected if there exists a directed path between any two vertices. A *strongly connected component* (SCC) is a subgraph $\mathcal{D}_S = (\mathcal{V}_S, \mathcal{E}_S)$ of $\mathcal{D}$ such that for every $u, v \in \mathcal{V}_S$ there exist paths from u to v and from v to u and is maximal with this property (i.e., any subgraph of $\mathcal{D}$ that strictly contains $\mathcal{D}_S$ is not strongly connected).

Definition 2 *An SCC is said to be linked if it has at least one incoming or outgoing edge from another SCC. In particular, an SCC is non-top linked if it has no incoming edges to its vertices from the vertices of another SCC.* ⋄

Furthermore, given a digraph $\mathcal{D} = (\mathcal{V}, \mathcal{E})$, we say that $\mathcal{D}$ is a *weakly connected digraph* if $(\mathcal{V}, \mathcal{E} \cup \mathcal{E}^{\intercal})$ is strongly connected, where $\mathcal{E}^{\intercal} \equiv \{(v', v) : (v, v') \in \mathcal{E}\}$

For any digraph $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ and any two sets $\mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{V}$ we define the *bipartite graph* $\mathcal{B}(\mathcal{S}_1, \mathcal{S}_2, \mathcal{E}_{\mathcal{S}_1,\mathcal{S}_2})$ where we call $\mathcal{S}_1$ the set of *left vertices*, and $\mathcal{S}_2$ the set of *right vertices*; and the edge set $\mathcal{E}_{\mathcal{S}_1,\mathcal{S}_2} = \mathcal{E} \cap (\mathcal{S}_1 \times \mathcal{S}_2)$. We call the bipartite graph $\mathcal{B}(\mathcal{V}, \mathcal{V}, \mathcal{E})$ the bipartite graph associated with $\mathcal{D}(\mathcal{V}, \mathcal{E})$. In the sequel we will make use of the *state bipartite graph*, $\mathcal{B}(\bar{A}) \equiv \mathcal{B}(\mathcal{X}, \mathcal{X}, \mathcal{E}_{\mathcal{X},\mathcal{X}})$, which is the bipartite graph associated associated with the state digraph $\mathcal{D}(\bar{A}) = (\mathcal{X}, \mathcal{E}_{\mathcal{X},\mathcal{X}})$, and the *system bipartite graph* $\mathcal{B}(\bar{A}, \bar{B}) = \mathcal{B}(\mathcal{U} \cup \mathcal{X}, \mathcal{X}, \mathcal{E}_{\mathcal{X},\mathcal{X}} \cup \mathcal{E}_{\mathcal{U},\mathcal{X}})$.

Given a bipartite graph $\mathcal{B}(\mathcal{S}_1, \mathcal{S}_2, \mathcal{E}_{\mathcal{S}_1,\mathcal{S}_2})$, a matching M corresponds to a subset of edges in $\mathcal{E}_{\mathcal{S}_1,\mathcal{S}_2}$ so that no two edges have a vertex in common, i.e., given edges $e = (s_1, s_2)$ and $e' = (s_1', s_2')$ with $s_1, s_1' \in \mathcal{S}_1$ and

$s_2, s_2' \in \mathcal{S}_2$, $e, e' \in M$ only if $s_1 \neq s_1'$ and $s_2 \neq s_2'$. Also, maximum matching M^* is a matching M that has the largest number of edges among all possible matchings.

Furthermore, it is possible to assign a weight to the edges in a bipartite graph, say $c(e)$ (where c is a function from $\mathcal{E}_{\mathcal{S}_1,\mathcal{S}_2}$ to $\mathbb{R}^+$). We thus obtain a *weighted bipartite graph*, and can introduce the concept of *minimum weight maximum assignment* problem. This problem consists in that of determining a maximum matching whose overall weight is as small as possible, i.e., a matching M^c such that

$$M^c = \arg \min_{M\in\mathcal{M}} \sum_{e\in M} c(e),$$

where $\mathcal{M}$ is the set of all maximum matchings. This problem can be efficiently solved using the Hungarian algorithm (Munkres, 1957), with complexity of $\mathcal{O}\left(\max\{|\mathcal{S}_1|, |\mathcal{S}_2|\}^3\right)$. We call the vertices in $\mathcal{S}_1$ and $\mathcal{S}_2$ belonging to an edge in M^*, the *matched vertices* with respect to (w.r.t.) M^*, otherwise, we call them *unmatched vertices*. It is worth noticing that there may exist more than one maximum matching. For ease of referencing, in the sequel, the term *right-unmatched vertices*, with respect to $\mathcal{B}(\mathcal{S}_1, \mathcal{S}_2, \mathcal{E}_{\mathcal{S}_1,\mathcal{S}_2})$ and a matching M, not necessarily maximum, will refer to those vertices in $\mathcal{S}_2$ that do not belong to an edge in M^*, dually a vertex from $\mathcal{S}_1$ that does not belong to an edge in M^* is called a *left-unmatched vertex*.

Now, we can interpret a maximum matching of a bipartite graph associated to a digraph, at the level of the digraph as follows (Pequito et al., 2016*a*).

Lemma 1 (Maximum Matching Decomposition) *Consider the digraph $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ and let M^* be a maximum matching associated with the bipartite graph $\mathcal{B}(\mathcal{V}, \mathcal{V}, \mathcal{E})$. Then, the digraph $\mathcal{D} = (\mathcal{V}, M^*)$ comprises a disjoint union of cycles and elementary paths (by definition an isolated vertex is regarded as an elementary path with no edges), beginning in the right-unmatched vertices and ending in the left-unmatched vertices of M^*, that span $\mathcal{D}$. Moreover, such a decomposition is* minimal*, in the sense that no other spanning subgraph decomposition of $\mathcal{D}(\bar{A})$ into elementary paths and cycles contains strictly fewer elementary paths.* ⋄

In addition, to make comparisons with previous work (namely, Rech and Perret (1991) and Li et al. (1996)), we need the following definition (Lin, 1974).

Definition 3 *Given a digraph $\mathcal{D}$, an elementary path in $\mathcal{D}$, also called a* stem*, is a cactus. Given a cactus $\mathcal{G} = (\mathcal{V}_\mathcal{G}, \mathcal{E}_\mathcal{G}) \subseteq \mathcal{D}$, and a cycle $\mathcal{C} = (\mathcal{V}_\mathcal{C}, \mathcal{E}_\mathcal{C}) \subseteq \mathcal{D}$, such that $\mathcal{G}$ and $\mathcal{C}$ have no vertices in common, and there is an edge from a vertex in $\mathcal{G}$ to a vertex in $\mathcal{C}$; then $\mathcal{G} \cup \mathcal{C} = (\mathcal{V}_\mathcal{G} \cup \mathcal{V}_\mathcal{C}, \mathcal{E}_\mathcal{G} \cup \mathcal{E}_\mathcal{C})$ is a cactus.* ⋄

Particularly, in the case where $\mathcal{D} = \mathcal{D}(\bar{A}, \bar{B})$, a cactus $\mathcal{G}$ in $\mathcal{D}$ is called an *input cactus* if the stem starts on an input vertex. Furthermore, we note that the decomposition in disjoint elementary paths and cycles, stated in Lemma 1, can be used to determine a spanning of the digraph in disjoint cacti (Pequito et al., 2016*a*).

When dealing with interconnected dynamical systems, the structure of the connection between the subsystems will create connections between the SCCs of different subsystem digraphs. This, in turn, makes it difficult to identify the SCCs of the system digraph of the overall system by analysing the SCCs of each subsystem digraph seperately and the connection to their neighbors. Hence, we introduce the concept of *reachability* (Dion et al., 2003). We thus say that a state vertex x in a system digraph is *input-reachable* or *input-reached* if there exists a path from an input vertex to it.

All of these constructions can be used to verify the structural controllability of an LTI system by analysing the associated graphs, as formally stated in the following result (Dion et al., 2003; Pequito et al., 2016*a*).

Theorem 1 *For LTI systems described by* (1)*, the following statements are equivalent:*

- (1) *The corresponding structured linear system $(\bar{A}, \bar{B})$ is structurally controllable;*
- (2) *The digraph $\mathcal{D}(\bar{A}, \bar{B})$ is spanned by a disjoint union of input cacti;*
- $(3i)$ *The non-top linked SCCs of the system digraph $\mathcal{D}(\bar{A}, \bar{B})$ are comprised of input vertices, and*
- $(3ii)$ *there is a matching of the system bipartite graph $\mathcal{B}(\bar{A}, \bar{B})$ without right-unmatched vertices;*
- $(4i)$ *Every state vertex is input-reachable, and*
- $(4ii)$ *there is a matching of the system bipartite graph $\mathcal{B}(\bar{A}, \bar{B})$ without right-unmatched vertices.* ⋄

4 Main Results

We begin this section by providing sufficient conditions for an interconnected dynamical system to be structurally controllable in the case where all the subsystems have the same structure (Theorem 2 and Theorem 3). We then focus on more general interconnected dynamical systems, called *serial systems*, and provide sufficient conditions for their structural controllability (Lemma 2), as well as an efficient distributed algorithm (Algorithm 1) to verify these conditions which has its correctness and complexity proven in Theorem 4. In light of these conditions, we explain why previous results in this line (Rech and Perret, 1991) presented conditions that are only sufficient instead of necessary and sufficient (Figure 2). Finally, we end this section by providing an efficient distributed algorithm (Algorithm 3) to verify the structural controllability of an arbitrary interconnected dynamical system, which has its correctness and complexity proven in Theorem 6. In order to perform this verification, each subsystem has to perform calculations using the information about itself and its neighbors. Furthermore, the subsystems must be able to communicate with each of their neighbors.

Often it is the case that the interconnected dynamical systems under analysis are comprised of subsystems that are similar among themselves. So, we begin by making this idea precise, and providing conditions to ensure structural controllability of such systems.

Definition 4 *Let $\bar{E} \in \{0,1\}^{r\times r}$, $\bar{A}', \bar{H} \in \{0,1\}^{n\times n}$, $\bar{B}' \in \{0,1\}^{n\times p}$, be matrices with the restriction that $\bar{E}_{i,i} = 0$ $(i = 1,\ldots,r)$. Then, we denote by $(\bar{A}', \bar{B}', \bar{H}, \bar{E}')$ the structural system $(\bar{A}, \bar{B})$, with $\bar{A} = (I_r \otimes \bar{A}') \vee (\bar{E} \otimes \bar{H})$ and $\bar{B} = I_r \otimes \bar{B}'$, where $\vee$ is the entry-wise logic* or*-operation. We call such a system* composed of r similar subsystems*, or a* similar system *for short.* ⋄

Remark 1 *Note that in the case of similar systems, $\bar{H}$ is the structural matrix modeling the interactions between each subsystem and its neighbors, all of which have the same structure.*

Definition 5 *Let $(\bar{A}, \bar{B})$ be the structural matrices associated with the interconnected dynamical system in* (1)*, we define the* condensed graph of the system *as being the digraph $\mathcal{D}^*(\bar{A}) \equiv \mathcal{D}(\mathcal{A}, \mathcal{E})$, where $a_i \in \mathcal{A} \equiv \{a_1,\ldots,a_r\}$ is a vertex representing the i–th subsystem, and $(a_i, a_j) \in \mathcal{E} \equiv \{(a_i,a_j)|E_{j,i} \neq 0\}$ a directed edge representing a communication from subsystem j to subsystem i. Moreover, if there is no directed edge ending in a vertex, this vertex is referred to as a* source. ⋄

Note that in the case that a system $(\bar{A}, \bar{B})$ is composed of r similar systems and parametrized by matrices $(\bar{A}', \bar{B}', \bar{H}, \bar{E})$ the condensed graph $\mathcal{D}^*(\bar{A})$ is the same as the digraph $\mathcal{D}(\bar{E})$. Now, we proceed to verify structural controllability of these systems when the subsystems are not structurally controllable by themselves.

Theorem 2 *Let the system $(\bar{A}, \bar{B})$ be composed of r similar components, and parametrized by $(\bar{A}', \bar{B}', \bar{H}', \bar{E})$, where $(\bar{A}', \bar{B}')$ is not structurally controllable, and $\mathcal{B}(\bar{A}', \bar{B})$ has a matching without right-unmatched vertices. The pair $(\bar{A}, \bar{B})$ is structurally controllable if and only if $(\bar{A}' \vee \bar{H}, \bar{B}')$ is structurally controllable and the condensed graph $\mathcal{D}^*(\bar{A})$ has no sources.* ⋄

Proof: To prove the equivalence, we begin by proving that the conditions are sufficient by contrapositive; subsequently, we prove directly that the conditions are also necessary.

Thus, we begin by noting that, since $(\bar{A}', \bar{B}')$ is not structurally controllable despite $\mathcal{B}(\bar{A}', \bar{B}')$ having a maximum matching without right-unmatched vertices, it follows, from Theorem 1–(4), that $\mathcal{D}(\bar{A}', \bar{B}')$ has a vertex which is not reachable from any input vertex. So assume that $\mathcal{D}^*(\bar{A})$ has a source, this implies that there is a subsystem $(\bar{A}'_j, \bar{B}'_j)$ with no incoming edges from other subsystems, and so the overall system digraph $\mathcal{D}(\bar{A}, \bar{B})$ has a state vertex without a path from any input vertex to it, and so, by Theorem 1–(4), $(\bar{A}, \bar{B})$ is not structurally controllable. Furthermore, $(\bar{A}' \vee \bar{H}, \bar{B}')$ is not structurally controllable. Since $\mathcal{B}(\bar{A}', \bar{B}')$ has a matching without right-unmatched vertices, so does $\mathcal{B}(\bar{A}' \vee \bar{H}, \bar{B}')$, which means, by Theorem 1–(4) that, $\mathcal{D}(\bar{A}' \vee \bar{H}, \bar{B}')$ must have a state vertex which is not reachable from any input vertex. This implies that the corresponding state vertex is not reachable from an input vertex in any of the subsystems (since a path from an input vertex in the overall system translates into one such path in $\mathcal{D}(\bar{A}' \vee \bar{H}, \bar{B}')$).

Finally, assume that $(\bar{A}' \vee \bar{H}, \bar{B}')$ is structurally controllable and that $\mathcal{D}^*(\bar{A})$ has no sources, then for each state vertex, there is a path from an input vertex to it, which implies, by Theorem 1–(4), that $(\bar{A}, \bar{B})$ is structurally controllable. ■

In the next result, we relax the assumptions from Theorem 2, about the structure of the dynamics of the subsystems. Thus, allowing for applications in the design of interconnections between subsystems that may fail to meet these criteria.

Theorem 3 *Given an interconnected dynamical system $(\bar{A}, \bar{B})$ composed of r similar components, and parametrized by $(\bar{A}', \bar{B}', \bar{H}, \bar{E})$, where $(\bar{A}', \bar{B}')$ is not structurally controllable, then $(\bar{A}, \bar{B})$ is structurally controllable if both $(\bar{A}' \vee \bar{H}, \bar{B}')$ is structurally controllable and $\mathcal{D}^*(\bar{A})$ is spanned by cycles.* ⋄

Proof: First, notice that if the digraph $\mathcal{D}^*(\bar{A})$ is spanned by cycles, every vertex in it is within a cycle, and in particular means that $\mathcal{D}^*(\bar{A})$ has no sources. Consequently, the method of proof of Theorem 2 is applicable to show that every state vertex has a path from an input vertex to it, so all that remains to show is that the $\mathcal{B}(\bar{A}, \bar{B})$ has no right-unmatched state vertices with respect to some maximum matching. To this end, we first assume (without loss of generality) that $\mathcal{D}^*(\bar{A})$ has one spanning cycle, and that the subsystems $(\bar{A}'_1, \bar{B}'_1), \cdots, (\bar{A}'_r, \bar{B}'_r)$ are ordered in such a way that $\bar{E}_{i+1,i} = 1$ for $i = 1,\ldots,r-1$, and $\bar{E}_{1,r} = 1$.

Now, denote the state and input vertices of the i–th subsystem by x^i_k with $k = 1,\cdots,n$ and u^i_l with $l = 1,\cdots,m$, respectively. In addition, let M be a maximum matching of $\mathcal{B}(\bar{A}' \vee \bar{H}, \bar{B}')$ without right-unmatched state vertices, then we can partition M' into three matchings M'_B, M'_A, M'_H comprising, respectively, the edges of M' of the form (u_l, x_k), those of the form (x_l, x_k) where $\bar{A}_{k,l} = 1$, and the remaining ones, that are of the form (x_l, x_k) where $\bar{A}_{k,l} = 0$ and $H_{k,l} = 1$. Finally, consider the matching M of $\mathcal{B}(\bar{A}, \bar{B})$ comprising the edges:

- (u^i_k, x^i_l), if (u_k, x_l) is in M'_B,
- (x^i_k, x^i_l), if (x_k, x_l) is in M'_A,
- (x^r_k, x^1_l), if (x_j, x_l) is in M'_H,
- (x^i_k, x^{i+1}_l), if (x_j, x_l) is in M'_H.

To show that this matching has no right-unmatched vertices, consider a state vertex x^i_k of $\mathcal{B}(\bar{A}, \bar{B})$, since M' has no right-unmatched vertices, x_k is not right-unmatched in $\mathcal{B}(\bar{A}', \bar{B}')$, and thus there is an edge (x_l, x_k) for some

l or $(u_{l'}, x_k)$ in M', but this implies that either (x_l^i, x_k^i), (x_l^{i-1}, x_k^i), or $(u_{l'}^i, x_k^i) \in M$ (where $i-1 = r$ when $i = 1$) from the construction above, thus $\mathcal{B}(\bar{A}, \bar{B})$ has no right-unmatched vertices w.r.t. M.

Finally, if more than one cycle is necessary to span the graph $\mathcal{D}^*(\bar{A})$, then the same argument applies to each cycle individually. ■

Remark 2 *The conditions in Theorem 3 are not necessary. Indeed, consider the example in Figure 1–(b) which is shown to be structurally controllable, yet the condensed graph is not spanned by cycles.* ◇

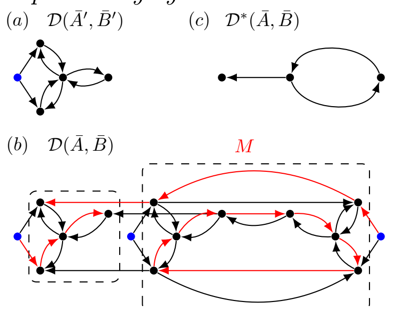

Fig. 1. In (a) we provide the digraph $\mathcal{D}(\bar{A}', \bar{B}')$ of a system that is not structurally controllable, where we represent in blue the single input vertex. By connecting three of these systems together as in (b) the system $(\bar{A}, \bar{B})$ becomes structurally controllable, as evidenced by the fact the matching M, associated to the path and cycle decomposition (see Lemma 1) depicted by the red edges, has no right-unmatched state vertices (since every state vertex has an incoming red edge), and the fact that every non-top linked SCC is comprised of an input vertex. Finally, in (c) we show that the condensed graph $\mathcal{D}^*(\bar{A})$ of the system in (b) is not spanned by cycles, showing that the condition in Theorem 3 is not necessary.

We now move toward methods of verifying structural controllability of interconnected dynamical systems by way of distributed algorithms. To this end, we begin by introducing a result that will allow us to infer structural controllability of a family of interconnected dynamical systems that we call serial systems. Later, we provide a computational method to perform this verification in a distributed manner, that is, while each subsystem only needs to have partial information about the system in order to verify if this is structurally controllable or not.

It is worth noting that in order to make it so that all of the algorithms in the present paper to work as intended, several assumptions have to be made about the subsystems, and how they communicate. Namely, that each subsystem has a processing unit, and can send arbitrary messages to its neighboring subsystems; in addition, each subsystem is aware of the number of subsystems in the overall system and possesses a unique *id*, and that the condensed graph of the system $\mathcal{D}^*(\bar{A})$ is weakly connected.

Lemma 2 *Consider the structural system $(\bar{A}, \bar{B})$ as in (1) with subsystems $(\bar{A}_1, \bar{B}_1), \ldots, (\bar{A}_r, \bar{B}_r)$. Then the system $(\bar{A}, \bar{B})$ is structurally controllable if there exist maximum matchings $M_0, \ldots, M_r$ of the bipartite graphs $\mathcal{B}(\bar{A}_1), \ldots, \mathcal{B}(\bar{A}_r)$ such that the following conditions hold:*

(1) For each subsystem $(\bar{A}_j, \bar{B}_j)$ with $j = 1, \ldots, r$, the non-top linked SCCs of $\mathcal{D}(\bar{A}_j, \bar{B}_j)$ is comprised of input vertices.

(2) the following bipartite graph admits a maximum matching without right-unmatched vertices
$$\mathcal{B}\left(\bigcup_{i=1}^{r} \mathcal{U}_L(M_i), \bigcup_{i=1}^{r} \mathcal{U}_R(M_i), \bigcup_{i=1}^{r} \bigcup_{j\neq i} \mathcal{E}_{\mathcal{U}_L(M_j), \mathcal{U}_R(M_i)}\right)$$
where $\mathcal{U}_L(M_i)$ and $\mathcal{U}_R(M_i)$ are the sets of left- and right-unmatched vertices, respectively, and $\mathcal{E}_{\mathcal{U}_L(M_j), \mathcal{U}_R(M_i)} \subseteq \mathcal{E}_{\mathcal{X},\mathcal{X}}$ is the set of edges from vertices in $\mathcal{U}_L(M_j)$ to vertices in $\mathcal{U}_R(M_i)$. ◇

Proof: First, note that the non-top linked SCCs of $\mathcal{D}(\bar{A}, \bar{B})$ are comprised of SCCs of the subsystem digraphs $\mathcal{D}(\bar{A}_i, \bar{B}_i)$, and that for one such SCC to be non-top linked, it must contain at least one non-top linked SCC of one of the $\mathcal{D}(\bar{A}_i, \bar{B}_i)$ in it. Therefore, since every non-top linked SCC of every $\mathcal{D}(\bar{A}_i, \bar{B}_i)$ is comprised of input vertices, and there are no edges from any neighboring system to input vertices, the non-top linked SCCs of $\mathcal{D}(\bar{A}, \bar{B})$ must be comprised of input vertices.

Secondly, note that the union of the maximum matchings M_i of the $\mathcal{B}(\bar{A}_i, \bar{B}_i)$ comprises a matching M of $\mathcal{B}(\bar{A}, \bar{B})$. Furthermore, let M' be the matching mentioned in condition (2). Since M' is comprised of edges from left-unmatched vertices to right-unmatched vertices of M, and so $M \cup M'$ is a matching of $\mathcal{B}(\bar{A}, \bar{B})$, and since by hypothesis the matching M' has no right-unmatched vertices, neither does $M \cup M'$. By Theorem 1–(3), this implies that the system is structurally controllable. ■

Note that using Lemma 2 we conclude that the system from Figure 2–(a) is structurally controllable, yet but using the characterization in Rech and Perret (1991), it is not possible to obtain the same conclusion. Furthermore, Lemma 2 provides only a sufficient condition for structural controllability. Nonetheless, these conditions can be verified in a distributed manner in the class of interconnected dynamical systems formally introduced next.

Definition 6 *We say that an interconnected dynamical system $(\bar{A}, \bar{B})$ as in (1) is a* serial system *if each vertex of the condensed graph $\mathcal{D}^*(\bar{A})$ has at most one outgoing edge.* ◇

Although serial systems seem a restrictive class of systems, they may exhibit a rich structure as exemplified in Figure 3. Furthermore, as stated before, serial systems enable us to verify the sufficient conditions for structural controllability in Lemma 2, in a distributed manner. Thus, in Algorithm 1 we present the procedure that

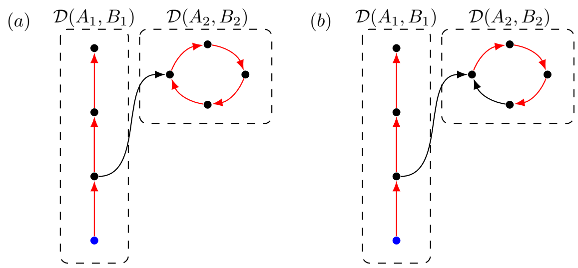

Fig. 2. In (a), we present the digraph associated to an interconnected dynamical system, where the different subsystems are represented inside the dashed boxes. Recall the definition of cactus, it can be readily be seen that the digraph $\mathcal{D}(\bar{A}, \bar{B})$ is spanned by the input cactus, depicted by the red edges, rendering the structural system $(\bar{A}, \bar{B})$ structurally controllable by Theorem 1–(2). In (b), however, we depict possible cacti that span each of the subsystem digraphs. Since a spanning cactus for $\mathcal{D}(\bar{A}_2, \bar{B}_2)$ has to include a stem comprising at least a vertex, neither of the cacti spanning $\mathcal{D}(\bar{A}_1, \bar{B}_1)$ and $\mathcal{D}(\bar{A}_2, \bar{B}_2)$ can contain any cycles, and there is no way of prolonging the stem that spans $\mathcal{D}(\bar{A}_1, \bar{B}_1)$ to include a stem spanning $\mathcal{D}(\bar{A}_2, \bar{B}_2)$. This shows that the conditions proposed in Rech and Perret (1991) are not necessary.

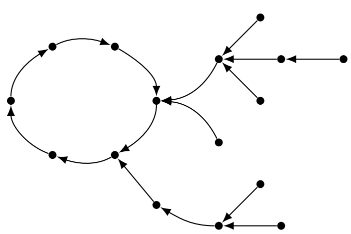

Fig. 3. Example of a possible condensed graph for a serial system, where each vertex represents a subsystem, and each directed edge a non-zero connection matrix, see Definition 6.

each agent deploys in order to verify the conditions in Lemma 2.

Before introducing Algorithm 1, we explain the functions that are used throughout the algorithm. All these functions should be able to be applied by the i–th subsystem, $\textsc{Send}(x, j)$ sends the value of the variable x to the j–th subsystem, while $\textsc{Rcv}(j)$ makes the system wait to receive a message from the j–th subsystem and subsequently reads this message. Note that both these functions can only be applied when systems i and j communicate with each other. For communication to be successful, we assume that the systems perform these steps synchronously (i.e. they wait for the responses of their neighbors). Finally, the procedure $\textsc{MinWtMaxMatch}(c, G)$ calculates the minimum weight maximum matching on the graph G using to this end, the cost function c, and boolean expressions contained in square brackets get evaluated (to **True** or **False**).

Remark 3 *Note that Algorithm 1 can be easily adapted to cover the case where each subsystem only has one incoming neighbor. In this case, instead of considering $\mathcal{B}_i$ as the bipartite graph associated to the i–th subsystem and all its incoming neighbors, we use the outgoing neighbors*

Algorithm 1
Distributed algorithm to verify sufficient conditions given in Lemma 2, for an arbitrary serial system.

procedure $\textsc{SeqStrtCtl}(\bar{A}_i, \bar{B}_i, r, \bar{E}_{i,j} \neq 0)$
$\triangleright$ r = total number of subsystems, $\bar{E}_{i,j}$ = connection matrices
  $\texttt{nghI}(i) \leftarrow \{j : \bar{E}_{i,j} \neq 0\}$
  $\texttt{nghO}(i) \leftarrow \{j : \bar{E}_{j,i} \neq 0\}$
  $\texttt{nghbs}(i) \leftarrow \texttt{nghI}(i) \cup \texttt{nghO}(i)$
  $\triangleright$ send the dynamic matrix to (the unique) outgoing neighbor
  for all $j \in \texttt{nghO}(i)$ **do**
    $\textsc{Send}(\bar{A}_i, j)$
  end for
  $\triangleright$ receive the dynamic matrix of the incoming neighbors $j \in \texttt{nghI}(i)$
  for all $j \in \texttt{nghI}(i)$ **do** $\triangleright$ $\texttt{nghI}(i) = \{j_1, \ldots, j_l\}$
    $\bar{A}_j \leftarrow \textsc{Rcv}(j)$
  end for

$$\bar{A}'_i \leftarrow \begin{bmatrix} \bar{A}_i & \bar{E}_{i,j_1} & \ldots & \bar{E}_{i,j_l} \\ 0 & \bar{A}_{j_1} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \bar{A}_{j_l} \end{bmatrix}$$

  $\bar{B}'_i \leftarrow \left[\bar{B}_i^\intercal, 0, \ldots, 0\right]^\intercal$
  $\mathcal{B}_i \leftarrow \mathcal{B}(\bar{A}'_i, \bar{B}'_i)$
  function $c_i(y, x_k,)$ $\triangleright$ define the weight function
    if $y = x_j$, and $k, j \leq n_i$ or $k, j > n_i$ **then**
      return 1
    else
      return 2
    end if
  end function
  $M_i \leftarrow \textsc{MinWtMaxMatch}(c_i, \mathcal{B}_i)$
  $\mathcal{U}_R(M_i) \leftarrow \{x_j : x_j$ right-unmatched w.r.t.M_i and $j \leq n_i\}$
  $\mathcal{N} \leftarrow \{$state vertices in non-top linked SCC of $\mathcal{D}(\bar{A}_i, \bar{B}_i)\}$
  $\texttt{mchd}(i) \leftarrow [\mathcal{U}_R(M_i) == \emptyset]$
  $\texttt{rchd}(i) \leftarrow [\mathcal{N} == \emptyset]$
  $\triangleright$ check if the whole system can be structurally controllable according to wether the conditions are satisfied in the current system or not
  $\texttt{ctld}(i) \leftarrow \texttt{rchd}(i) \wedge \texttt{mchd}(i)$
  for $k = 1, \ldots, r$ **do**
    for all $j \in \texttt{nghbs}(i)$ **do**
      $\textsc{Send}(\texttt{ctld}(i), j)$
      $\texttt{ctld}(j) \leftarrow \textsc{Rcv}(j)$
    end for
    $\triangleright$ reconsider the answer in light of the values from the neighbors current answer
    $\texttt{ctld}(i) \leftarrow \texttt{ctld}(i) \bigwedge_{j \in \texttt{nghbs}(i)} \texttt{ctld}(j)$
  end for
  $\triangleright$ return **True** if the system is structurally controllable and **False** otherwise
  return $\texttt{ctld}(i)$
end procedure

instead. ◇

The next result concerns the correctness and complexity of Algorithm 1.

Theorem 4 *Algorithm 1 is correct, i.e., it verifies the sufficient conditions given in Lemma 2 for an arbitrary serial system. Moreover, it has computational complexity* $\mathcal{O}\left(\max_{i=1,\ldots,r} N_i^3\right)$*, with* $N_i = m_i + \sum_{j\in\mathcal{I}_i\cup\{i\}} n_j$*, where* m_i *and* n_i *are the dimensions of the input and state space for the* i*–th subsystem, and* $\mathcal{I}_i \subseteq \{1,\ldots,r\}$ *is the set of subsystems which output to the* i*–th subsystem.* ◇

Proof: To prove the correctness of Algorithm 1, we start by proving the claim that a minimum weight maximum matching M of $\mathcal{B}_i$ w.r.t. the weight-function c_i defined in step 14 induces maximum matchings on $\mathcal{B}(\bar{A}_i)$, as well as on $\mathcal{B}(\bar{A}_j)$ for any subsystem $(\bar{A}_j, \bar{B}_j)$ with nonzero connection matrix $\bar{E}_{i,j}$: let M_i be the matching resulting from restricting M to the edges of $\mathcal{B}(\bar{A}_i)$, and in order to derive a contradiction, assume that M_i is not a maximum matching of $\mathcal{B}(\bar{A}_i)$. As a direct consequence of Berge's theorem (see for example Theorem 1 in Berge (1957)) the set of right-unmatched vertices of any matching contains the right-unmatched vertices of some maximum matching, so let M_i' be a maximum matching such that $\mathcal{U}_R(M_i') \subseteq \mathcal{U}_R(M_i)$. Furthermore, let $S_i \subseteq M$ be the set of edges from a vertex not in $\mathcal{X}_i$ to a vertex in $\mathcal{X}_i$, and let S_i' be those edges in S_i that end in some vertex in $\mathcal{U}_R(M_i')$. Now, $(M \setminus (M_i \cup S_i)) \cup M_i' \cup S_i'$ is a matching of $\mathcal{B}_i$ with the same number of edges as M (since it has the same number of right-unmatched vertices) and with an overall weight lower than that of M (since by hypothesis $S_i' \subsetneq S_i$), which contradicts the fact that M is a minimum weight maximum matching. The same argument works for the matching M_j of $\mathcal{B}(\bar{A}_j)$ with $j \neq i$, replacing left-unmatched vertices with right-unmatched vertices.

Now, since $(\bar{A}, \bar{B})$ is a serial system, there is at most one $k \neq i$ with nonzero matrix $E_{k,i}$. Therefore, we let M' and M'' be the maximum matchings of $\mathcal{B}(\bar{A}_i)$ resulting from the maximum matchings of $\mathcal{B}_i$ and $\mathcal{B}_k$, respectively. Then, by Lemma 4 in Pequito et al. (2016*a*), there exists a maximum matching M that has as left-unmatched vertices those of M'' and as right-unmatched vertices those of M'. Subsequently, we only need to check for each subsystem that there is a minimum weight maximum matching of $\mathcal{B}_i$ (w.r.t. the weight function w_i) that has no right-unmatched state vertices. Thus, in Algorithm 1, we set up the necessary structures until step 14.

Now, in step 19 the i–th subsystem computes the maximum matching M_i of $\mathcal{B}_i$, and in step 20 the system calculates the associated set of right-unmatched vertices. Next, in step 21 the subsystem calculates the set of state vertices in a non-top linked SCC of $\mathcal{D}(\bar{A}_i, \bar{B}_i)$, and in steps 22 and 23, it verifies the existence of right-unmatched state vertices of the i–th subsystem w.r.t. the matching M_i, and the existence of in a non-top linked SCC of $\mathcal{D}(\bar{A}_i, \bar{B}_i)$. Finally the subsystem decides if the whole system is structurally controllable or not in steps 24–29. More precisely, after an initial guess has been made and stored in `ctld`(i), the subsystem updates this variable with the corresponding variable of its neighbors, and repeats this r times. Note that after k iterations of the steps 26–29 the subsystem has updated `ctld`(i) with the corresponding values of all subsystems at k edges of distance from it. Since the condensed graph of the systems is weakly connected, the communication between subsystems is undirected, and there are only r subsystems, `ctld`(i) = **True** if and only if all subsystems had initially `ctld`(j) = **True**. Finally, in step 30 the subsystem returns the value **True** or **False** depending on wether or not the system satisfies the conditions of Lemma 2.

Lastly, the complexity of Algorithm 1 is computed as follows: since all of the steps have linear complexity except determining the minimum weight maximum matching of $\mathcal{B}_i$ in step 19, for which the Hungarian algorithm can be used with complexity $\mathcal{O}\left(|N_i|^3\right)$, with $N_i = p_i + \sum_{j\in\mathcal{I}_i\cup\{i\}} n_j$, $\bar{A}_i \in \{0,1\}^{n_i\times n_i}$ and $\bar{B}_i \in \{0,1\}^{n_i\times p_i}$ and $\mathcal{I}_i \subseteq \{1,\ldots,r\}$ is the set of indexes of subsystems incoming to the i–th subsystem Munkres (1957). This procedure has to be applied to each of the r subsystems, which implies that the complexity of the algorithm becomes $\mathcal{O}\left(\max_{i=1,\ldots,r} N_i^3\right)$. ■

Remark 4 *Note that if the system were not serial then there could be a subsystem, with* k*–th system that outputs to both the* i*– and* j*–th subsystems. This could mean that when computing maximum matchings of* $\mathcal{B}_i$ *and* $\mathcal{B}_j$ *separately we could match the state vertex of the* k*–th subsystem to two different state vertices, one of the* i*–th subsystem and one of the* j*–th subsystem. Furthermore, note that if there is a subsystem with incoming edges from every other system, the algorithm will calculate a maximum matching in a centralized manner.*

Now we move toward distributed algorithms that are able to verify structural controllability of interconnected dynamical systems at large. In this case, each subsystem is required to share only partial information about its structure with its neighbors. This algorithm, however, has a higher computational complexity than Algorithm 1. In order to infer structural controllability, we employ Theorem 1–(4), and begin by presenting an algorithm to verify if each of the state vertices in the digraph associated to an interconnected dynamical system as in (1) has a path from an input vertex to it.

Theorem 5 *Algorithm 2 is correct (i.e., it returns* ***True*** *if and only if every state vertex in the* i*–th subsystem digraph is input-reached). Furthermore, Algorithm 2 has complexity* $\mathcal{O}\left(\max\left\{r^2, Nr, N\max_{i=1,\ldots,r} n_i\right\}\right)$*, where* n_i *is*

Algorithm 2

Distributed algorithm to verify condition $(4i)$ of Theorem 1.

1: **procedure** REACHED($\bar{A}_i, \bar{B}_i, \bar{E}_{i,k} \neq 0, \bar{E}_{k,i} \neq 0, r$)
2: $\quad$ $\texttt{nghI}(i) \leftarrow \{j : \bar{E}_{i,j} \neq 0\}$
3: $\quad$ $\texttt{nghO}(i) \leftarrow \{j : \bar{E}_{j,i} \neq 0\}$
4: $\quad$ $\texttt{nghbs}(i) \leftarrow \texttt{nghI}(i) \cup \texttt{nghO}(i)$
5: $\quad$ $N_i \leftarrow \#\{\text{SCCs of } \mathcal{D}(\bar{A}_i)\}$
6: $\quad$ $\texttt{SCCs}(i) \leftarrow \{(i, N_i)\}$
$\quad$ ▷ the subsystems communicate with each other to learn how many SCCs each subsystem has, in order to find the necessary number of communication steps
7: $\quad$ **for** $k = 1, \ldots, r$ **do**
8: $\quad\quad$ **for all** $j \in \texttt{nghbs}(i)$ **do**
9: $\quad\quad\quad$ SEND($\texttt{SCCs}(i), j$)
10: $\quad\quad\quad$ $\texttt{SCCs}(j) \leftarrow$ RCV(j)
11: $\quad\quad\quad$ $\texttt{SCCs}(i) \leftarrow \texttt{SCCs}(i) \cup \texttt{SCCs}(j)$
12: $\quad\quad$ **end for**
13: $\quad$ **end for**
14: $\quad$ $N \leftarrow \sum_{j=1}^{r} \texttt{SCCs}(j)$
15: $\quad$ $\texttt{rchd}(i) \leftarrow \{\}$ ▷ list of input-reached vertices
$\quad$ ▷ add the vertices with incoming edges from input vertices
16: $\quad$ **for** $j = 1, \ldots, n_i$ **do**
17: $\quad\quad$ **if** $\exists k : (\bar{B}_i)_{j,k} = 1$ **then**
18: $\quad\quad\quad$ ADDTO($x_j, \texttt{rchd}(i)$)
19: $\quad\quad$ **end if**
20: $\quad$ **end for**
21: $\quad$ **for** $k = 1, \ldots, N$ **do**
$\quad\quad$ ▷ transmit to outgoing neighbors which vertices that communicate with them have been input reached
22: $\quad\quad$ **for all** $j \in \texttt{nghO}(i)$ **do**
$\quad\quad\quad$ ▷ $M_{\bullet,l}$ is the l–th column of M
23: $\quad\quad\quad$ SEND($\{x_l : (i, x_l) \in \texttt{rchd}(i) \text{ and } (\bar{E}_{j,i})_{\bullet,l} \neq 0\}, j$)
24: $\quad\quad$ **end for**
$\quad\quad$ ▷ add vertices reached from the neighbors' input reached vertices
25: $\quad\quad$ **for all** $j \in \texttt{nghI}(i)$ **do**
26: $\quad\quad\quad$ $\texttt{avail}(j) \leftarrow$ RCV(j)
27: $\quad\quad\quad$ $\texttt{rchd}(i) \leftarrow \texttt{rchd}(i) \cup \{x_t : (\bar{E}_{i,j})_{t,l} = 1, x_l \in \texttt{avail}(j)\}$
28: $\quad\quad$ **end for**
$\quad\quad$ ▷ verify which vertices are reachable from the inputs by using k edges between subsystems steps
29: $\quad\quad$ **for** $l = 1, \ldots, n_i$ **do**
30: $\quad\quad\quad$ $\texttt{rchd}(i) \leftarrow \texttt{rchd}(i) \cup \{x_t : (\bar{A}_i)_{t,s} = 1, x_s \in \texttt{rchd}(i)\}$
31: $\quad\quad$ **end for**
32: $\quad$ **end for**
$\quad$ ▷ return **True** if every state vertex the i–th subsystem digraph is input-reached, and **False** otherwise
33: $\quad$ **return** $[\#\texttt{rchd}(i) == n_i]$
34: **end procedure**

the dimension of the state space of the i–th subsystem, and $N = \sum_{i=1}^{r} n_i$, where k_i is the number of SCCs in the i–th subsystem digraph.

◇

Proof: Note that, since each subsystem can establish two-way communication with its neighbors, the communication graph is strongly connected, and thus the instructions in steps 7–13 only need to be executed (at most) r times in order to receive all pairs (id, #SCCs) in the system. Subsequently the total number N of SCCs can be computed in step 14.

Now, assume that each subsystem has a strongly connected state digraph $\mathcal{D}(\bar{A}_i)$. Then, if the system has an input vertex, i.e., if $\bar{B}_i \neq 0$, each of the state vertices of the i–th subsystem is added to $\texttt{rchd}(i)$ in the first iteration of the for-loop in steps 21–31, namely in the for-loop 29–31. Furthermore, note that in the case where each subsystem has a strongly connected system digraph, $N = r$, and a path from an input vertex to a state vertex contains at most r edges between different subsystem digraphs. Therefore, in this case, after N iterations of steps 21–31 all vertices that may be reached by a path from an input vertex have been added to $\texttt{rchd}(i)$.

Alternatively, if the i–th subsystem is not strongly connected, then, assume, without loss of generality, that $\bar{A}_i$ is a block matrix, with submatrices $\bar{A}_i^1, \ldots, \bar{A}_i^l$ along the diagonal so that $\mathcal{D}(\bar{A}_i^1), \ldots, \mathcal{D}(\bar{A}_i^l)$ are strongly connected. Furthermore, let $\bar{B}_i^1, \ldots, \bar{B}_i^l$ be the restriction of $\bar{B}_i$ to the lines in used by $\bar{A}_i^1, \ldots, \bar{A}_i^l$ respectively. Then, consider the interconnected dynamical comprising, instead of the i–th subsystem $(\bar{A}_i, \bar{B}_i)$, the subsystems $(\bar{A}_i^1, \bar{B}_i^1), \ldots, (\bar{A}_i^l, \bar{B}_i^l)$ connected amongst them and to other subsystems according to $\bar{A}_i$. By applying this procedure to every subsystem whose state digraph is not strongly connected, we obtain an interconnected dynamical system, where each subsystem has a strongly connected digraph. Note also, that we didn't change the state digraph of the overall system, thus a state vertex in the overall system digraph is input-reached if and only if it was input-reached in the original system digraph. Now, since this the number of SCCs in all subsystems of the original system digraph of this system is N subsystems, from the previous paragraph we conclude that after N iterations of steps 21–31, every state vertex in $\mathcal{D}(\bar{A}_i)$ that is input-reached in $\mathcal{D}(\bar{A}, \bar{B})$ has been added to $\texttt{rchd}(i)$.

Thus, we have proven that for any interconnected dynamical system, every state vertex of $\mathcal{D}(\bar{A}_i)$ has a path from some input vertex in the overall system if and only if $\#\texttt{rchd}(i) = n_i$.

Finally, we analyze the complexity of Algorithm 2. We begin by noting that the SCCs of $\mathcal{D}(\bar{A}_i)$ can be computed in $\mathcal{O}(n_i)$. Now, each of the steps in the for-loop 7–13 can be executed in constant time, which implies that the for-loop incurs in complexity $\mathcal{O}(r\#\texttt{nghbs}(i))$ which is bounded by $\mathcal{O}(r^2)$. Furthermore, the steps 22–24 and 25–28 can be executed in constant complexity, thus these loops incur in complexity $\mathcal{O}(\#\texttt{nghO}(i))$ and $\mathcal{O}(\#\texttt{nghI}(i))$, respectively. Finally, the for-loop in steps 29–31, incurrs in linear complexity (on the number, n_i, of state variables). So in conclusion, the complexity of Algorithm 2 becomes $\mathcal{O}\left(\max\left\{r^2, Nr, N \max_{i=1,\ldots,r} n_i\right\}\right)$. ■

Next, we present a distributed algorithm to verify structural controllability when the subsystems only have access to neighboring subsystems. Briefly, the algorithm consists in verifying both conditions ($4i$) and ($4ii$) of Theorem 1 in a distributed manner. Condition ($4i$) of Theorem 1, can be verified by applying Algorithm 2. On the other hand, Theorem 1–($4ii$) requires one to compute a maximum matching in a distributed manner. This can be achieved by reducing the problem of finding a maximum matching to that of computing a maximum flow (Ahuja et al., 1993). However, since we only need to detect the existence of right-unmatched vertices, we only need to compute a maximum *preflow* (which corresponds to a flow, where the flow on the incoming edges need not be equal to the flow on the outgoing edges of each vertex). To this end, we employ the distributed algorithm provided in Shekhovtsov and Hlaváč (2013). In order to achieve this reduction, one first takes the overall system bipartite graph and provides an orientation to each edge, from left-vertex to right-vertex; then one adds two extra vertices, called *source* and *sink*; finally one adds an edge from the source to each of the left-vertices of the bipartite graph, and from each of the right-vertices to the sink and assigns to each vertex a *capacity* of 1 (Ahuja et al., 1993). The computation of the maximum flow is then done distributedly, where each subsystem works to maximize the flow from the source to the sink within a *region* of the graph comprising the subsystems bipartite graph, the source and the sink (note that the source and sink lie in all regions, which does not impair the distribution of the algorithm, since the systems need not keep track of the excess on the source or the sink), and any vertices in other subsystems to which the system is connected. This is achieved through a *push-relabel* algorithm, briefly described as follows: each of the vertices in a region keeps track of an *excess* (which corresponds to the difference between the incoming and outgoing flow), and a *label* or *height*. The excess is then pushed from higher labels to lower labels increasing the flow through the edges between them until it reaches the sink, or the boundary. Once this is achieved, the excess accumulated in the boundary is passed to the corresponding neighboring region, and the iterations begin again. However, the existence of boundary vertices limits the parallelization, as two instantiations of the algorithm can only (in general) be computed simultaneously, if the regions do not share vertices other than the source or the sink.

From this point onwards we refer to the individual instances of the *parallell region discharge* algorithm presented in Shekhovtsov and Hlaváč (2013) as PRD. Furthermore, we assume PRD considers the following parameters: the digraph on which it operates, the capacity function, and the neighbors with which it shares vertices other than the source or the sink. Also, PRD returns a maximal preflow on the digraph.

Algorithm 3
Distributed algorithm to verify condition (4) of Theorem 1.

procedure CONTROLLED($\bar{A}_i, \bar{B}_i, \bar{E}_{i,k} \neq 0, \bar{E}_{k,i} \neq 0, r$)
  $\texttt{nghI}(i) \leftarrow \{j : \bar{E}_{i,j} \neq 0\}$
  $\texttt{nghO}(i) \leftarrow \{j : \bar{E}_{j,i} \neq 0\}$
  $\texttt{nghbs}(i) \leftarrow \texttt{nghI}(i) \cup \texttt{nghO}(i)$
  ▷ verify if every state vertex is input-reached by deploying Algorithm 2
  $\texttt{rchd}(i) \leftarrow$ REACHED($\bar{A}_i, \bar{B}_i, \bar{E}_{i,k} \neq 0, \bar{E}_{k,i} \neq 0, r$)
  ▷ we set up the graph for applying the Parallel region discharge, where s and t correspond to the source and sink, respectively, the x and u vertices correspond to state and input vertices. The upper index i is the index of the subsystem they belong to, and the upper index R and L indicate if they are right or left vertices
  $\mathcal{V}_i \leftarrow \{s,t\} \cup \{x_k^{i,L}, x_k^{i,R}\}_{k=1}^{n_i} \cup \{u_k^i\}_{k=1}^{m_i}$
  $\mathcal{E}_{i,i} \leftarrow \{(x_j^{i,L}, x_{j'}^{i,R}) : (\bar{A}_i)_{j',j} = 1\} \cup \{(u_j^i, x_{j'}^{i,R}) : (\bar{B}_i)_{j',j} = 1\}$
  for all $j \in \texttt{nghbs}(i)$ **do**
    $\mathcal{V}_{i,j} \leftarrow \{x_k^{j,R} : (\bar{E}_{j,i})_{k,\bullet} \neq 0\}$
    $\mathcal{E}_{i,j} \leftarrow \{(x_l^{i,L}, x_k^{j,R}) : (\bar{E}_{j,i})_{k,l} = 1\}$
    $\mathcal{V}_{j,i} \leftarrow \{x_k^{j,L} : (\bar{E}_{j,i})_{\bullet,k} \neq 0\}$
    $\mathcal{E}_{j,i} \leftarrow \{(x_k^{j,L}, x_l^{i,R}) : (\bar{E}_{i,j})_{l,k} = 1\}$
  end for
  $\mathcal{E}_s \leftarrow \{s\} \times \{x_k^{i,L}\}_{k=1}^{n_i}$
  $\mathcal{E}_t \leftarrow \{x_k^{i,R}\}_{k=1}^{n_i} \times \{t\}$
  $\mathcal{E} \leftarrow \mathcal{E}_s \cup \mathcal{E}_t \cup \mathcal{E}_{i,i} \cup \bigcup_{j \in \texttt{nghbs}(i)} (\mathcal{E}_{j,i} \cup \mathcal{E}_{i,j})$
  $\mathcal{V} \leftarrow \mathcal{V}_i \cup \bigcup_{j \in \texttt{nghbs}(i)} (\mathcal{V}_{j,i} \cup \mathcal{V}_{i,j})$
  $\mathcal{D} \leftarrow (\mathcal{V}, \mathcal{E})$
  function $c(e \in \mathcal{E}_i)$
    return 1 ▷ all edges have unitary capacity
  end function
  ▷ Deploy a Parallel Region Discharge algorithm to obtain a preflow f on $\mathcal{D}$, with capacity function c
  $f \leftarrow \text{PRD}(\mathcal{D}, c, \texttt{nghbs(i)})$
  $\texttt{mchd}(i) \leftarrow [\sum_{e \in \mathcal{E}_t} f(e) == n_i]$
  ▷ check if the whole system can be structurally controllable according to wether the conditions are satisfied in the current system or not
  $\texttt{ctld}(i) \leftarrow \texttt{rchd}(i) \wedge \texttt{mchd}(i)$
  for $k = 1, \ldots, r$ **do**
    ▷ reconsider the controllability of the overall system, in light of the data received from the neighbors
    for all $j \in \texttt{nghbs}(i)$ **do**
      SEND($\texttt{ctld}(i), j$)
      $\texttt{ctld}(j) \leftarrow$ RCV(j)
      $\texttt{ctld}(i) \leftarrow \texttt{ctld}(i) \wedge \texttt{ctld}(j)$
    end for
  end for
  return $\texttt{ctld}(i)$
  ▷ return **True** if the system is structurally controllable and **False** otherwise
end procedure

Theorem 6 *Algorithm 3 is correct, i.e., it verifies* (4) *of Theorem 1. Furthermore, it has a computational complexity of*

$$\mathcal{O}\left(\max\{r^2, Nr, N \max_{i=1,\ldots,r} n_i, r\beta^2 \max_{i=1,\ldots,r} n_i^3\}\right)$$

where β is the number of boundary vertices, and the re-

maining variables are the same as described in Theorem 5. ⋄

Proof: In order to verify the correctness of Algorithm 3, we have to check if both conditions ($4i$) and ($4ii$) of Theorem 1 are verified. Furthermore, in order to perform this verification in a distributed manner, each subsystem must verify that all vertices in its digraph are input-reached in $\mathcal{D}(\bar{A}, \bar{B})$, which is done by employing Algorithm 2 in step 5; and that none of its state vertices are right-unmatched w.r.t. some maximum matching of $\mathcal{B}(\bar{A}, \bar{B})$. Once this has been achieved, it was already argued in the proof of Theorem 4, that the for loop in steps 25–31 determines if these conditions are violated in any of the subsystems.

Now, to verify that Algorithm 3 determines if there are right-unmatched vertices in the i–th subsystem, we note that in steps 6–21 we generate the digraph $\mathcal{D}$ comprising the right- and left-vertices of the i–th subsystem bipartite graph, and the boundary vertices of the i–th region according to the precepts in Shekhovtsov and Hlaváč (2013). Once the digraph $\mathcal{D}$ is computed, we apply PRD to it in step 22, thus obtaining a preflow from source to sink on $\mathcal{D}$ which is maximum amongst preflows on the whole graph. By the guarantees provided in Shekhovtsov and Hlaváč (2013), together with the equivalence between the maximum matching and maximum flow problems, presented in Ahuja et al. (1993) we guarantee that $\sum_{e\in\mathcal{E}_t} f(e)$ is equal to the number of right-matched vertices in a maximum matching of the system bipartite graph, that are state vertices of the i–th subsystem. So, by comparing $\sum_{e\in\mathcal{E}_t} f(e)$ with n_i in step 23, we are able to infer if there are right-unmatched vertices in the i–th subsystem w.r.t. some maximum matching $\mathcal{B}(\bar{A}, \bar{B})$. Thus the algorithm returns **True** if and only if every state vertex of the system digraph is input-reached, and there are no right unmatched vertices in the system bipartite graph.

Now, since all of the steps of the algorithm have linear complexity except for step 5 and step 22, the complexity of Algorithm 3 is given by the maximum of these. Knowing that step 5 has a complexity of $\mathcal{O}\left(\max\left\{r^2, Nr, N \max_{i=1,\dots,r} n_i\right\}\right)$ (see Theorem 6, where N is the number of SCCs on each of the subsystems), all that remains to infer is the complexity of step 22. This algorithm, as described in Shekhovtsov and Hlaváč (2013), iteratively performs a push-relabel procedure, followed by pushing the accumulated excess in the boundary to a neighboring system. The push-relabel algorithm has complexity $\mathcal{O}\left(n_i^3\right)$ (Ahuja et al., 1997; Goldberg, 2008), and the necessary iterations of the region discharge that each subsystem has to complete, can be bounded by β^2 where β is the number of boundary vertices in the whole system bipartite graph (that is, the number of vertices in the bipartite graph that have to be shared by several subsystems). Also, in the worst-case scenario where each of the subsystems is connected to every other subsystem, the region discharge steps have to be executed sequentially. So, the complexity of step 22 is given by $\mathcal{O}\left(r\beta^2 \max_{i=1,\dots,r} n_i^3\right)$, resulting in an overall complexity of $\mathcal{O}\left(\max\left\{r^2, Nr, N \max_{i=1,\dots,r}\left\{n_i, r\beta^2 n_i^3\right\}\right\}\right)$. ■

Remark 5 *The computational complexity of Algorithm 4 is dominated by the PRD, since Algorithm 2, which is required to assess reachability in a distributed fashion, has lower computational complexity. In particular, if the network is connected, then* $n_i > 0$ *and* $r\beta^2 n_i^3 > n_i$ *for each subsystem* i*, where* $0 < r \leq \beta$*. Subsequently, the computational complexity of the PRD algorithm reduces to* $\mathcal{O}\left(Nr\beta^2 \max_{i=1,\dots,r} n_i^3\right)$. ⋄

5 Illustrative Examples

Now, we provide a small example of how Algorithm 3 (and Algorithm 2, which is required as a subroutine) runs on the interconnected dynamical system depicted in Figure 4–1. In particular, we aim to emphasize the distributed nature of Algorithm 3.

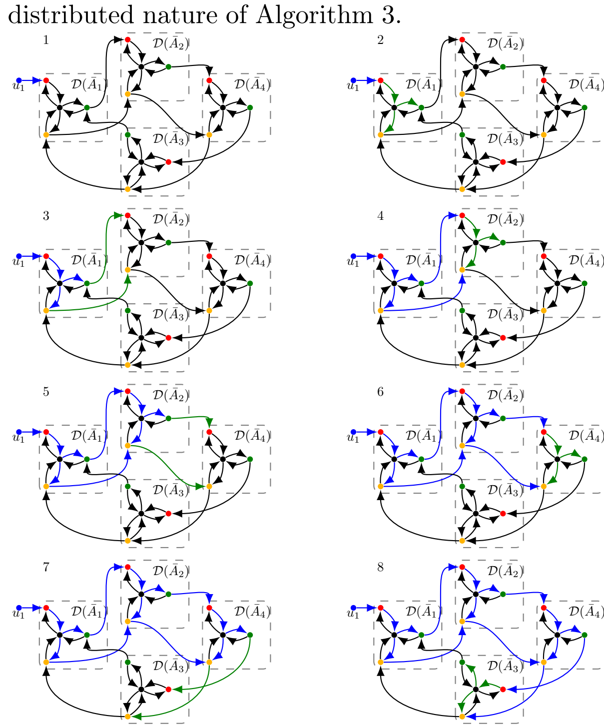

Fig. 4. Example of the procedure of Algorithm 2 applied to the system digraph presented in Figure 4–1, comprising 4 different subsystems (depicted inside of the dashed boxes), only one of which has an input edge (labeled u_1). In each subfigure, the blue edges represent those that comprised a path from an input vertex, and the green edges denote those that were added in this iteration or communication step of the algorithm. Finally, the even-labeled subfigures correspond to an iteration of Algorithm 2, and the odd-labeled ones correspond to a communication step between subsystems.

In Figure 4–1, we present the digraph associated to an interconnected dynamical system comprising four subsystems. Since only subsystem $(\bar{A}_1, \bar{B}_1)$ has an input vertex, it readily follows from Theorem 1 that none of the other subsystems can be structurally controllable. Now, we employ Algorithm 3 to verify the structural controllability of the interconnected dynamical system. After the initialization steps are completed, we use Algorithm 2, the iterations of which can be seen in Figure 4–2 to Figure 4–8: in each iteration (even-labeled subfigures) new vertices are seen to be input-reached (the targets of the green edges), and in each communication step (odd-labeled subfigures) the subsystems communicate to its outgoing neighbors which of their vertices are reached after the iteration has been completed. As can be seen in Figure 4–8, all vertices have been reached after four iterations, which in this case, corresponds to the number of SCCs in all subsystems.

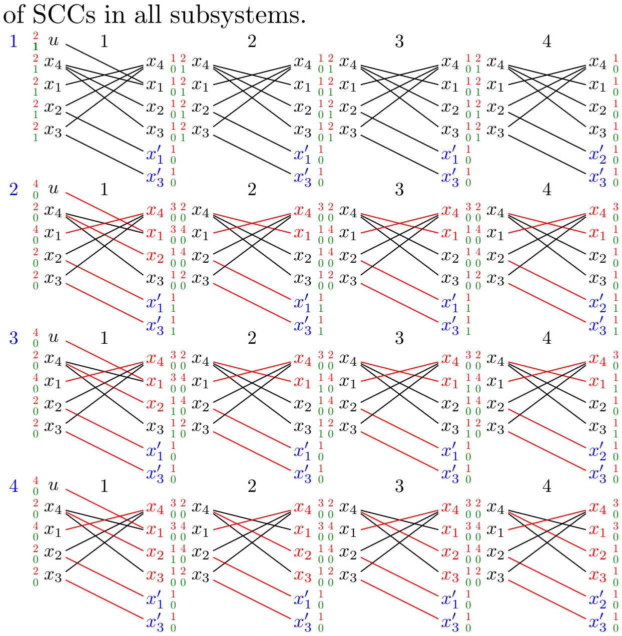

Fig. 5. Example of the PRD algorithm applied to the verification of structural controllability of the system presented in Figure 4–1. For convenience of referencing, the vertices are given labels rather than colors (x_4 being the black vertex in each subsystem and the others can be easily inferred). At the left of each left-vertex and at the right of each right-vertex we insert two numbers, the one in green represents the excess of the corresponding vertex, whereas the red number represents its label. Edges in red represent those where the capacity has been saturated; and right-vertices in red represent the ones for which the edge to the sink has been saturated. Finally, the vertices in blue represent vertices that belong to other regions, i.e., boundary vertices. In order to simplify, in this, we do not include boundary vertices from incoming subsystems. Note also, that in this instance, we can run the algorithm in all regions simultanously, since by not considering the incoming edges from other regions in the region graph, we do not allow for flow to be sent back through these edges.

In Figure 5, we consider an example of a run of the region discharge algorithm running on the bipartite graph associated to the digraph in Figure 4–1. In this example, we begin applying PRD in step 1 by initializing the labels at 2 for each left-vertex, and at 1 for each right-vertex; we also saturate all edges from the source, which makes it so that all of the left-vertices start with an excess of 1. By successive pushing and relabeling, we reach the configuration in 2 where all the excess has either been pushed to the sink (and thus the corresponding right-vertex is presented in red) or to the boundary of the region. In step 3, we discharge the excess from the boundary into the adjacent regions so that, for example, the right-vertex x_3 in each of the regions has now an excess of 1. Finally, by applying push-relabel again in each of the regions, we reach step 4 where all the edges from right-vertices to the sink have been saturated (and are thus displayed in red) showing that there is a maximum pre-flow saturating all edges to the sink, and equivalently that there is a maximum matching with no right-unmatched vertices. So, in combination with the analysis of Figure 4 we conclude that the interconnected dynamical associated to the digraph system in Figure 4–1 is structurally controllable.

Remark 6 *The computational complexity of solving the maximum-flow in a centralized versus the distributed version implemented by the PRD, whose implementations are available in Shekhovtsov and Hlaváč (2011*a*) and discussed in detail in Shekhovtsov and Hlaváč (2011*b*). More specifically, it is provided a trade-off between the CPU time required and the number of nodes and the number of subsystems with the same number of nodes. In particular, the PRD takes in average twice the computational time required by the centralized algorithm to solve the maximum-flow. Furthermore, we notice that Algorithm 2 amounts to a depth-first search, which performance is essentially the same as the distributed verification algorithm proposed.* ◇

6 Conclusions and Further Research

In this paper, we have provided several necessary and/or sufficient conditions to verify structural controllability for interconnected linear time-invariant dynamical systems based on the local information accessible to each subsystem. Subsequently, we have provided distributed and efficient (i.e., polynomial in dimension of the state and input) algorithms to verify a necessary and sufficient condition for structural controllability. The results presented readily extend to discrete time-invariant interconnected dynamical systems, since the controllability criterion stays the same. Furthermore, by duality between controllability and observability the results also apply to structural observability verification of discrete/continuous linear time-invariant interconnected dynamical systems. Whereas the results presented pertain to verify structural conditions, it would be of interest to address design problems; for instance, which state variables need to be actuated, or which inputs should be used, to ensure a given structural property. On the other hand, it would be of interest to understand if the conditions provided could be adapted to the case where some of the entries in the structure of the subsystems and their interconnections are known exactly (which corresponds to the case where only some of the components of the overall system are assumed to be reliable). Ultimately, such an extension would shed light on the relationship between structural and non-structural system-theoretic properties; hence, leading to a better understanding of the resilience and performance of interconnected dynamical systems.

7 Acknowledgements

The first author would like to thank ACCESS Linnaeus Center, for their hospitality, as much of the writing of this paper and the research that led to it were carried out while visiting KTH in the Spring of 2014.

References

Ahuja, R. K., Kodialam, M., Mishra, A. K. and Orlin, J. B. (1997), 'Computational investigations of maximum flow algorithms', *European Journal of Operational Research* **97**(3), 509–542.

Ahuja, R. K., Magnanti, T. L. and Orlin, J. B. (1993), *Network flows: theory, algorithms, and applications*, Prentice Hall.

Anderson, B. D. O. and Hong, H.-M. (1982), 'Structural controllability and matrix nets', *International Journal of Control* **35**(3), 397–416.

Berge, C. (1957), 'Two theorems in graph theory', *Proceedings of the National Academy of Sciences of the United States of America* **43**(9), 842.

Blackhall, L. and Hill, D. J. (2010), 'On the structural controllability of networks of linear systems', *2nd IFAC Workshop on Distributed Estimation and Control in Networked Systems* pp. 245–250.

Chen, C. and Desoer, C. (1967), 'Controlability and observability of composite systems', *IEEE Transactions on Automatic Control* **12**(4), 402–409.

Cormen, T. H., Stein, C., Rivest, R. L. and Leiserson, C. E. (2001), *Introduction to Algorithms*, 2nd edn, McGraw-Hill Higher Education.

Davison, E. (1977), 'Connectability and structural controllability of composite systems', *Automatica* **13**(2), 109 – 123.

Davison, E. and Wang, S. (1975), 'New results on the controllability and observability of general composite systems', *IEEE Transactions on Automatic Control* **20**(1), 123–128.

Davison, E. and Özgüner, Ü. (1983), 'Characterizations of decentralized fixed modes for interconnected systems', *Automatica* **19**(2), 169 – 182.

Dion, J.-M., Commault, C. and der Woude, J. V. (2003), 'Generic properties and control of linear structured systems: a survey.', *Automatica* pp. 1125–1144.

Goldberg, A. V. (2008), The partial augment-relabel algorithm for the maximum flow problem, *in* D. Halperin and K. Mehlhorn, eds, 'Algorithms - ESA 2008', number 5193 *in* 'Lecture Notes in Computer Science', Springer Berlin Heidelberg, pp. 466–477.

Hespanha, J. P. (2009), *Linear Systems Theory*, Princeton Press, Princeton, New Jersey.

Li, K., Xi, Y. and Zhang, Z. (1996), 'G-cactus and new results on structural controllability of composite systems', *International Journal of Systems Science* **27**(12), 1313–1326.

Lin, C. (1974), 'Structural controllability', *IEEE Transactions on Automatic Control* (3), 201–208.

Munkres, J. (1957), 'Algorithms for the assignment and transportation problems', *Journal of the Society of Industrial and Applied Mathematics* **5**(1), 32 – 38.

Pequito, S., Kar, S. and Aguiar, A. P. (2015), 'On the complexity of the constrained input selection problem for structural linear systems', *Automatica* **62**, 193 – 199.

Pequito, S., Kar, S. and Aguiar, A. P. (2016*a*), 'A framework for structural input/output and control configuration selection of large-scale systems', *IEEE Transactions on Automatic Control* **61**(2), 303–318.

Pequito, S., Kar, S. and Aguiar, A. P. (2016*b*), 'Minimum cost input/output design for large-scale linear structural systems', *Automatica* **68**, 384 – 391.

Rech, C. and Perret, R. (1991), 'About structural controllability of interconnected dynamical systems', *Automatica* **27**(5), 877 – 881.

Shekhovtsov, A. and Hlaváč, V. (2011*a*), 'A distributed algorithm for mincut/maxflow combining path augmentation'. `http://cmp.felk.cvut.cz/~shekhovt/d_maxflow/`

Shekhovtsov, A. and Hlaváč, V. (2011*b*), A Distributed Mincut/Maxflow Algorithm Cmobining Path Augmentation and Push-Relabel, Research Report K333–43/11, Center for Machine Perception, Czech Technical University in Prague, Prague, Czech Republic.

Shekhovtsov, A. and Hlaváč, V. (2013), 'A distributed mincut/maxflow algorithm combining path augmentation and push-relabel', *International Journal of Computer Vision* **104**, 315–342.

Shields, R. W. and Pearson, J. B. (1976), 'Structural controllability of multi-input linear systems', *IEEE Transactions Automatic Control* **AC-21**(3).

Wang, S. and Davison, E. (1973), 'On the controllability and observability of composite systems', *IEEE Transactions on Automatic Control* **18**(1), 74–74.

Wolovich, W. and Hwang, H. (1974), 'Composite system controllability and observability', *Automatica* **10**(2), 209 – 212.

Yang, G.-H. and Zhang, S.-Y. (1995), 'Structural properties of large-scale systems possessing similar structures', *Automatica* **31**(7), 1011 – 1017.

Yonemura, Y. and Ito, M. (1972), 'Controllability of composite systems of tandem connection', *IEEE Transactions on Automatic Control* **17**(5), 722–724.

Zhou, T. (2015), 'On the controllability and observability of networked dynamic systems', *Automatica* **52**(0), 63 – 75.

Özgüner, Ü. and Hemani, H. (1985), 'Decentralized control of interconnected physical systems', *International Journal of Control* **41**(6), 1445–1459.