



HAL
open science

Structural controllability of networks with dynamical nodes

Christian Commault

► **To cite this version:**

Christian Commault. Structural controllability of networks with dynamical nodes. 2018. hal-01806913

HAL Id: hal-01806913

<https://hal.univ-grenoble-alpes.fr/hal-01806913v1>

Preprint submitted on 4 Jun 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Structural controllability of networks with dynamical nodes

Christian Commault^a

^a*Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, 38000 Grenoble, France
Email:christian.commault@gipsa-lab.grenoble-inp.fr*

Abstract

In this paper we consider networks which are governed by external controls. The internal nodes of the network are composed of linear dynamic systems, in such a way that the network graph is a static structure, while the life of the system is located in the nodes. Moreover, we assume that each node is a linear structured system with a single ingoing state vertex and a single outgoing state vertex. Recall that structured systems are dynamic systems where the entries of the corresponding state space matrices are independent parameters. To such structured systems, one can associate a directed graph that allows to study a lot of generic properties of the system. Combining the network graph with the node graphs provides with a global graph which will allow the study of the structural controllability of the global system. We give necessary and sufficient controllability conditions for the global system, with respect to the controllability of the nodes and to the properties of the network graph. The conditions are expressed in terms of an auxiliary graph which has the same size as the network graph, but takes into account important features of the node graphs. These conditions are particularly simple in the case where all the nodes have the same structure, and therefore the same graph. Besides its interest in terms of insight in the controllability of networks with dynamical nodes, this approach is also much more efficient computationally than a direct checking of the controllability of the global graph.

Keywords: Network controllability, structured system theory, graph theory.

1. Introduction

1.1. Context

Networks and communications are present at almost any instant of our life. They may correspond to real devices as in power networks or railway networks, but also to virtual links as in social networks. They show up both in large scale communications as in satellite networks and in the microscopic world as in blood cells or in brain connections. All fields of science (physics, biology, computer science, engineering) are impacted by these communication issues. As a result,

in the last decades, emerged a scientific network community, who developed which is now called *Network Science* [1]. In the last ten years, the network community showed a real, and increasing, interest in the controllability issues [2, 3]. Some very famous papers contributed to spread out the concepts and tools of control theory in a very wide scientific audience [4, 5]. These papers, and those who followed, make use of the graph interpretation of controllability in so-called structured systems, and apply it to networks. This application was questioned in [6], where it was argued that the graph of a network is mainly a static interconnection structure while the dynamics are hidden in the nodes. Notice that some recent works consider also the possibility of having dynamic edges [7, 8].

On another hand, in control theory, there has been a constant interest in the control of so-called composite systems. The study of such composite systems is an old field of research in control theory [9, 10, 11], which is still very active [12, 13, 14, 15] and related to the notion of network of network [16]. Although networks and composite systems may lead to similar models, they correspond to different philosophies. In network theory, the main object of study is the network scheme itself, which means that the living part of the network which is contained in the nodes, should be as simple as possible.

1.2. Contributions

With the latter remark in mind, the present paper is an attempt to complete the ideas of [6] by considering that each node of the network is made of linear dynamic system with single ingoing state and single outgoing state. Since the corresponding dynamic models have generally loosely defined entries, we assume that we deal with parameterized systems, called structured systems. A structured system is a usual state space linear representation, where all the entries of the corresponding matrices are zeros or free parameters. To such a structured system, one can associate a graph whose vertices correspond to the variables, and whose edges represent their interactions, *i.e.* correspond to matrices' non-zero entries. The graph allows to get simple conditions for some generic properties of the system, generic properties being properties that are true for almost any value of the parameters [17, 18, 19]. Recall that the structural controllability of a system can be checked through two sub-properties of the associated graph [17]:

1. a control connection condition, which means that each state vertex can be reached from an input,
2. a so-called no-dilation condition, which corresponds to a full generic rank for the composite matrix $[A, B]$.

In this paper, we will indeed work with three graphs.

- The directed graph of the network, with control nodes describing external signals acting on the network, and internal nodes which are connected according to the interconnection scheme. We call it the *network graph*.

- A directed graph of each node corresponding to its structured model, with a unique ingoing vertex and a unique outgoing vertex. We call it the *node graph*.
- The graph obtained from the network graph by expanding each node according to the corresponding node graph. We call it the *global graph*.

Our aim will be to study the controllability of the global system/graph with respect to the properties of the network graph and of the node graphs.

The main contributions of the paper are the following:

- The definition of a simple model of a network with dynamical nodes, for which each node is defined by a graph with a single *in-vertex* (a unique state vertex receiving the signals coming from the control and from the other nodes) and a single *out-vertex* (which value is the only information that can be transmitted from this node to the other nodes).
- The definition of node graphs having specific properties, those which can be covered by a disjoint set of cycles (called *CYC*), and those which can be covered by a disjoint in-out path and a set of cycles (called *IOP*).
- The definition of an auxiliary graph which is simply constructed from the network graph using the properties *CYC* and/or *IOP* of the corresponding node graphs.
- Very simple and intuitive necessary and sufficient conditions for structural controllability of the global graph in terms of the auxiliary graph. This characterization leads to a noticeable gain in terms of complexity for the controllability analysis, compared with a direct application of the structural controllability conditions on the global graph.
- The possibility to extend the famous Minimal Controllability Problem to this framework.

In this paper there is a deliberate choice to start with a very simple (but not trivial) model for the node dynamics in order to get controllability conditions which are simple, easy to interpret and to check algorithmically.

1.3. Comparison with related works

The literature on networks with dynamical nodes is mainly divided in two parts, depending on the nature of the model chosen for the nodes. This model is either a classical one, *i.e.* with fixed state space matrices, or a structured model. The advantages and limitations of these two models have been discussed in a lot of papers, see for example [19]. These two points of view should not be opposed, but rather be seen as complementary. Our choice of a structured system model is based on the desire to have a complete graph framework and to get an insight in the system not related to particular values of the entries of the state space model matrices.

As mentioned in Subsection 1.1, there is a plentiful of literature dealing with the controllability of interconnected linear systems defined by usual state space interconnected models. They were first called composite systems [9, 10] and are now more often called networks [20, 13, 14] or systems of systems [16]. The controllability of these systems is closely related with the eigenstructure of the components, therefore their analysis is mainly based on the Popov-Belevich-Hautus (PBH) controllability criterion. Most of these works consider very general interconnection schemes, consequently, they give only either necessary or sufficient conditions, or very hard to check controllability conditions. As an illustration, in a recent paper [21], the authors use node systems with one-dimensional communications, as in the present paper. It is interesting to note that, although they assume that all nodes are identical, the obtained necessary and sufficient controllability conditions remain complex and difficult to interpret.

When the components of the network are supposed to be structured systems, and then associated with a graph, the analysis was generally based on a characterization of controllability by particular graph objects called *cacti* [17]. The covering of the graph by cacti is another way to test the structural controllability. The problem is then formulated as: given components that are covered by cacti, how must be the connections in the network in order to cover the global graph by cacti [11, 12] or generalized cacti [22]? The difficulty comes from the fact that there is no systematic way to characterize all the cacti covering a given graph. As a consequence, these papers only provide with sufficient conditions for structural controllability. In a recent paper [15], the authors refine these sufficient conditions, in particular for so-called serial systems. Moreover, they put a particular emphasis on possible distributed computations.

The outline of this paper is as follows. In Section 2, we state the controllability problem for networks with dynamic nodes. In Section 3, structured systems are recalled, together with some known results on structural controllability. In Section 4, some preliminary simple results and observations are presented. In Section 5, we study the case where all the nodes have the same structure, and therefore the same graph. The general case is treated in Section 6 where we provide with a necessary and sufficient controllability condition. In this Section, we also examine how the Minimal Controllability Problems can be dealt with in our context. In Section 7 we provide with a complexity analysis which shows the efficiency of our method. In Section 8, a brief conclusion ends the paper.

2. Problem statement

2.1. The network and his graph

We consider a *network* \mathcal{N} , composed of N *internal nodes* communicating through directed links. A set of m *control nodes* allows to inject some external scalar signals into the network to control his behaviour. The whole network can be represented by an directed graph $G(\mathcal{N}) = (V_{\mathcal{N}}, E_{\mathcal{N}})$, called the *network graph*. The vertex set is $V_{\mathcal{N}} = \{z_1, \dots, z_N\} \cup \{u_1, \dots, u_m\}$, where the z_i 's represent the internal nodes and the u_i 's represent the control nodes. The edge

set represents the directed connections between the nodes. The network graph indeed corresponds to the static structure of the network. In the following, for simplicity, but with an abuse of notation, we will call $u_i(t)$ the value of the control signal delivered by node u_i . The context will avoid ambiguity. The network graph will be given edge weights to quantify the interactions between the nodes of the network in a way which will be made precise later.

2.2. The node dynamic system

The node z_i is associated with a dynamic system Σ_i , which is a SISO linear state space model with n_i states, a scalar control γ_i and scalar output y_i . The control acts on a given state x_{ik} called *in-state*. A particular state x_{ij} , called *out-state*, transmits his value to the other nodes according to the network graph. The system is represented as

$$\Sigma_i : \begin{cases} \dot{x}_i(t) = A_i x_i(t) + B_i \gamma_i(t) \\ y_i(t) = C_i x_i(t). \end{cases} \quad (1)$$

A_i is a $n_i \times n_i$ matrix. B_i is a column vector with all entries at zero except for the k -th entry which is 1, and corresponds with the in-state x_{ik} . C_i is a row vector with all entries at zero except for the j -th entry which is 1, and corresponds with the out-state x_{ij} .

2.3. The connection scheme

Composing the network structure with the state space model in each node results in a *global system* Σ of state dimension $n = \sum_{i=1}^N n_i$. The control signal for the system Σ_i is the weighted combination of network control signals and output node signals in accordance with the network graph.

$$\gamma_i(t) = \sum_{j=1}^N w_{ij} y_j(t) + \sum_{l=1}^m w'_{il} u_l(t), \quad (2)$$

where w_{ij} is the weight of the link from node z_j to node z_i , w'_{il} is the weight of the link from control node u_l to node z_i , these weights are zero when there is no edge in the network graph between the corresponding nodes. Recall that, from equation (1), $y_j(t)$ is indeed the value of the out-state of node z_j . In summary, the states of the global system Σ are the states of the different node systems, and the controls of the global system are the controls of the network.

The aim of this paper is the study of the controllability of the global system Σ , with respect to the controllability of the node systems Σ_i and to the structure of the network appearing in the network graph $G(\mathcal{N})$. We will assume that the node systems are structured, in the sense that only their zero/non-zero structure is known. The context of our study will then be the structured system framework that we recall now.

3. Structured systems

3.1. Linear structured systems

We consider a linear system with parameterized entries and denoted by Σ_Λ .

$$\Sigma_\Lambda : \dot{x}(t) = A_\Lambda x(t) + B_\Lambda u(t), \quad (3)$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^m$ the control vector, and A_Λ and B_Λ are matrices of appropriate dimensions. The system is called a linear structured system if the entries of the composite matrix $J_\Lambda = [A_\Lambda, B_\Lambda]$ are either fixed zeros or independent parameters [18, 19]. The vector $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_k)^\top \in \mathbb{R}^k$, with $^\top$ indicating transposition, denotes the vector of independent parameters of the composite matrix J_Λ .

For linear structured systems one can study generic properties. *i.e.*, properties which are true for almost all values of the parameters collected in Λ . More precisely a property is said to be *generic* (or *structural*) if it is true for all values of the parameter vector Λ , outside a proper algebraic variety in the parameter space \mathbb{R}^k . Recall that a proper algebraic variety is the zero set of some non-trivial polynomial with real coefficients in the k parameters of the system. A proper algebraic variety has Lebesgue measure zero.

For a structured matrix M_Λ , the rank of M_Λ for almost any value of Λ , in the previous sense, is called its generic rank and denoted as $g\text{-rank } M_\Lambda$. Notice that $g\text{-rank } M_\Lambda$ is also the maximum value of $\text{rank } M_\Lambda$ for any value of Λ , for more details see [18].

A directed graph $G(\Sigma_\Lambda) = (V, E)$ can be associated with the linear structured system Σ_Λ in Equation (3):

- the vertex set is $V = X \cup U$, where X and U are the state and control vertex sets given by $\{x_1, x_2, \dots, x_n\}$ and $\{u_1, u_2, \dots, u_m\}$, respectively,
- the edge set is $E = \{(x_i, x_j) | A_{\Lambda ji} \neq 0\} \cup \{(u_i, x_j) | B_{\Lambda ji} \neq 0\}$, where $A_{\Lambda ji}$ (resp. $B_{\Lambda ji}$) denotes the entry (j, i) of the matrix A_Λ (resp. B_Λ).

Recall that a path in $G(\Sigma_\Lambda)$ from a vertex v_{i_0} to a vertex v_{i_q} , is a sequence of edges $(v_{i_0}, v_{i_1}), (v_{i_1}, v_{i_2}), \dots, (v_{i_{q-1}}, v_{i_q})$, such that $v_{i_t} \in V$ for $t = 0, 1, \dots, q$ and $(v_{i_{t-1}}, v_{i_t}) \in E$ for $t = 1, 2, \dots, q$. The vertices v_1, \dots, v_p are then said to be *covered* by the path. If $v_{i_0} \in U$ and, $v_{i_q} \in X$, the path is called a control-state path. A path for which $v_{i_0} = v_{i_q}$ is called a circuit. A *stem* is a control-state path which does not meet twice the same vertex. A system is said to be *control-connected* if any state vertex is the end vertex of a stem. A cycle is a circuit which does not meet twice the same vertex, except for the initial/end vertex. Two paths are disjoint when they cover disjoint sets of vertices. When some stems and cycles are mutually disjoint, they constitute a *disjoint set of stems and cycles*.

The notion of structural controllability was introduced by Lin, who proved a necessary and sufficient condition for structural controllability of single-control systems in terms of graph-theoretic objects called cacti, see [17]. The following result can be proved to be equivalent to Lin's result, see for instance [23, 24].

Theorem 1. *Let Σ_Λ be the linear structured system defined by (3) with associated graph $G(\Sigma_\Lambda)$. The system is structurally controllable if and only if*

- *The system Σ_Λ is control-connected,*
- *The state vertices of $G(\Sigma_\Lambda)$ can be covered by a disjoint set of stems and cycles.*

The verification of the first condition can be done using a simple labelling algorithm [25]. The second condition is less intuitive and a bit more complex to check numerically, this will be detailed in the next Subsection.

3.2. Efficient test of the second controllability condition

The second condition of Theorem 1 is indeed equivalent to $g\text{-rank}[A_\Lambda, B_\Lambda] = n$. It is also equivalent to the existence of a maximum matching of size n in the bipartite graph associated with the matrix $[A_\Lambda, B_\Lambda]$, which provides with an efficient way to check the condition. Let us now be more precise.

The bipartite graph $H(\Sigma_\Lambda) = (H^+, H^-; W)$ associated with Σ_Λ (or equivalently associated with $G(\Sigma_\Lambda)$) is defined as follows. The vertex set H^+ is given by $X^+ \cup U$, the vertex set H^- is given by X^- , with $X^+ = \{x_1^+, \dots, x_n^+\}$ the first set of state vertices, $X^- = \{x_1^-, \dots, x_n^-\}$ the second set of state vertices and $U = \{u_1, \dots, u_m\}$ the set of input vertices. The edge set is $W = W_A \cup W_B$ with $W_A = \{(x_j^+, x_i^-) | A_{\Lambda ij} \neq 0\}$ and $W_B = \{(u_j, x_i^-) | B_{\Lambda ij} \neq 0\}$. In the former, for instance $A_{\Lambda ij} \neq 0$ means that the (i, j) -th entry of the matrix A_Λ is a parameter (structurally non-zero).

A *matching* in a bipartite graph $H(\Sigma_\Lambda) = (H^+, H^-; W)$ is an edge set $M \subseteq W$ such that the edges in M have no common vertex. A matching M is called maximum if its cardinality is maximum, in general, a maximum matching is not unique. The maximum matching problem is the problem of finding such a matching of maximal cardinality. This is a classical problem in combinatorial optimisation, for which polynomial algorithms are available, see [26] for a classical reference and [27, 28] for more recent and high-performance methods. The following result summarizes some well known results, see for example [29, 18].

Proposition 1. *Let Σ_Λ be the linear structured system defined by (3) with associated bipartite graph $H(\Sigma_\Lambda)$. The generic rank of $[A_\Lambda, B_\Lambda]$ is equal to the cardinality of a maximal matching in $H(\Sigma_\Lambda)$.*

As a result, the following three properties are equivalent:

- *The state vertices of $G(\Sigma_\Lambda)$ can be covered by a disjoint set of stems and cycles.*
- *The generic rank of $[A_\Lambda, B_\Lambda]$ is n .*
- *The cardinality of a maximum matching in $H(\Sigma_\Lambda)$ is n .*

Occasionally, in the following we will also be interested in the dynamic system without control, therefore only in the A_Λ matrix. With the matrix A_Λ we can

associate the graph $G(A_\Lambda)$ and the bipartite graph $H(A_\Lambda)$, these graphs being sub-graphs of $G(\Sigma_\Lambda)$ and $H(\Sigma_\Lambda)$ respectively. We can then get the following Corollary.

Corollary 1. *Let $H(A_\Lambda)$ be the bipartite graph associated with the A_Λ matrix. The following three properties are equivalent:*

- *The state vertices of $G(A_\Lambda)$ can be covered by a disjoint set of cycles.*
- *The generic rank of A_Λ is n .*
- *The size of a maximum matching in $H(A_\Lambda)$ is n .*

3.3. Application to the network context

Let us now come back to the networks with dynamical nodes introduced in Section 2. When the system Σ_i of node z_i is structured, it can be associated with a node graph $G(\Sigma_{\Lambda,i})$. In $G(\Sigma_{\Lambda,i})$, the vertex corresponding to the in-state (out-state) is called the in-vertex (out-vertex). A path from the in-vertex to the out-vertex is called an *in-out path*. We call *internal node cycle*, a cycle which covers vertices belonging to $G(\Sigma_{\Lambda,i})$. To study the controllability of a node $G(\Sigma_{\Lambda,i})$, we consider that the state vertices are associated with a dummy input $\gamma_i(t)$ as in Equation 1 which is connected with the in-state vertex.

Assume that the node systems are structured systems with independent parameters, and that the weight of links in the network graph which appear in Equation (2) are also independent parameters. The global system Σ_Λ obtained from the network graph, the node systems of type (1) and the connection Equation (2), is then a structured system. The parameter set Λ of this global system is the union of parameter sets of the node systems and of the weights w_{ij} and w'_{il} of Equation (2) considered as parameters. The graph of this global system is $G(\Sigma_\Lambda) = (V, E)$. This graph can easily be obtained from $G(\mathcal{N})$ and from the graphs $G(\Sigma_{\Lambda,i})$ associated with the node systems, as described in Subsection 3.1. The vertex set V of $G(\Sigma_\Lambda)$ is composed of the state vertices of the node graphs and of the control nodes of $G(\mathcal{N})$. For consistency, in the global graph the control nodes will be called control vertices. The edge set E of $G(\Sigma_\Lambda)$ contains the edges of node graphs $G(\Sigma_{\Lambda,i})$. Moreover, for each edge (z_i, z_j) of $G(\mathcal{N})$, there is an edge in $G(\Sigma_\Lambda)$, from the out-vertex of $G(\Sigma_{\Lambda,i})$ to the in-vertex of $G(\Sigma_{\Lambda,j})$ and for each edge (u_k, z_l) of $G(\mathcal{N})$, there is an edge from the control vertex u_k to the in-vertex of $G(\Sigma_{\Lambda,l})$.

From this point, we have a graph associated with the global structured system, therefore the controllability of Σ_Λ could be studied thanks to Theorem 1. However, $G(\Sigma_\Lambda)$ is generally high-dimensional, and it is then of interest to study its controllability using properties of the nodes and of the network graph.

3.4. About the terminology used in this paper

- In classical graph theory, the terms vertex and node are used interchangeably. In this paper they are used with distinct meanings. Nodes are associated with the network graph and are in some sense "super vertices".

Vertices are associated with the node graphs and then with the global graph. In figures, nodes will be represented by large circles and vertices by smaller ones.

- Here, in and out vertices are indeed distinguished state vertices of a given node. They correspond to the only ingoing and outgoing possible communications with the other nodes. They should not be confused with inputs and outputs which appear in most of works in control theory, where inputs and outputs are external variables. Consequently, the in-out paths as they appear here are different from input-output paths which are used for solving a lot of classical control problems for structured systems [19].
- As mentioned before, the network graph (and the to be defined later auxiliary network graph) does not correspond to a dynamical system and therefore speaking about its controllability does not make sense. However, we will need to check the two conditions of Theorem 1 on the network graph. The notions which were defined for the graph of a structured dynamical system (stem, control-connection, cycles) will also be used in a similar way for the graph $G(\mathcal{N})$. In particular, we will say that the network graph is control connected if there is path from a control node to any internal node. In order to check the second condition of Theorem 1, we will also associate a bipartite graph with the auxiliary graph as in Subsection 3.2.

Example 1. *The construction of the global graph can be illustrated on the following example. The network has five nodes and two controls. The graph of the network, the graph of the node systems and the resulting global graph are given in Figure 1. The in-vertices of the node graphs are in green, the out-vertices of the node graphs are in red and the control vertices of the network graph are in blue. Notice that, in order to lighten the figures, we do not name the vertices of the node graphs.*

Remark 1. *In practice, we will be interested in large scale networks, say with hundreds or thousands of nodes, while the node graphs will be of a modest dimension, often with less than ten vertices. A particular, and interesting, case is when a node contains only one state. Then the in-state (in-vertex) and out-state (out-vertex) are the same state (vertex).*

4. Preliminary results

From the definition of the global graph and from Theorem 1, some simple results can be established on the structural controllability of the global system.

Proposition 2. *The global system is control connected if and only if the network graph and the node graphs are control connected*

Proof:

The simple proof is left to the reader. ■

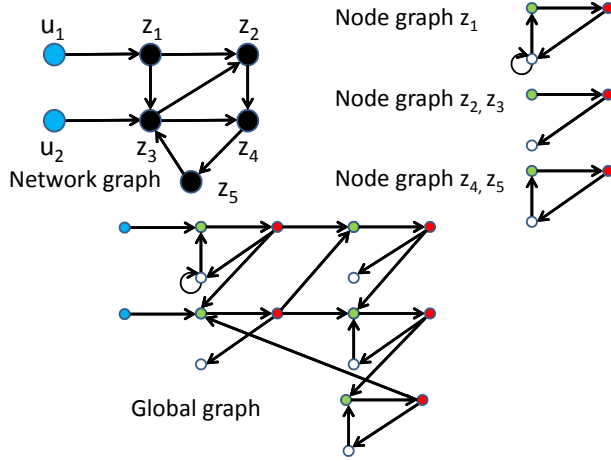


Figure 1: Example 1

Proposition 3. *The global system is structurally controllable only if the node systems are structurally controllable.*

Proof:

Assume that the global system is structurally controllable. First, from Proposition 2, the node graphs $G(\Sigma_{\Lambda,i})$ must be control connected. Moreover, the state vertices of the global graph $G(\Sigma_{\Lambda})$ can be covered by a disjoint set of stems and cycles, see Theorem 1. From the construction of the global graph, these stems and cycles can be categorized as follows:

- Internal node cycles.
- Cycles resulting of cycles of the network graph, in which case the cycle contains an in-out path in the graph of the nodes it passes through. In such a node, the vertices of the node which are not on the in-out path must be covered by internal cycles.
- Stems of the global graph. When it passes through a node, the stem contains an in-out path in the node graph. For the terminal node, the global stem contains an in-state vertex path in the node graph. In both cases, the vertices of the node which are not on the in-out path or stem must be covered by internal cycles.

To summarize, a disjoint set of stems and cycles covering all state vertices in the global graph $G(\Sigma_{\Lambda})$ induces a disjoint set of stems and cycles covering all state vertices in each node graph $G(\Sigma_{\Lambda,i})$. Therefore, the structural controllability of each node graph is a necessary condition for the structural controllability of the

global graph. ■

The previous result may seem to be very natural. However, using more sophisticated connection schemes, the global system may be controllable while some subsystems are uncontrollable in the usual sense. But, in such cases, the node controllability condition shows up provided *virtual inputs* corresponding to the connecting inputs are added to the usual controls [11, 12, 15].

Proposition 4. *The global system is structurally controllable if the node graphs and the network graph are control connected, and the state vertices of each node graph can be covered by a disjoint set of internal node cycles.*

Proof:

From Proposition 2, the control connection for the node graphs and for the network graph, imply the control connection of the global graph.

The collection of the disjoint cycles covering all the state vertices in the node graphs $G(\Sigma_{\Lambda,i})$ constitutes a set of disjoint cycles covering all the state vertices in the global graph $G(\Sigma_{\Lambda})$. So, the global graph satisfies the conditions of Lin's Theorem and the global system is structurally controllable. ■

Remark 2. *To complete Proposition 4, it appears that the second condition of Theorem 1 need not be satisfied by the network graph, to get the global controllability. This is illustrated by the example in Figure 2. In this example, the two nodes have the same graph where all vertices can be covered by a cycle. It can be seen that the global graph is structurally controllable while the network graph does not satisfy condition 2 of Theorem 1, indeed the network graph is a dilation in the sense of [17].*

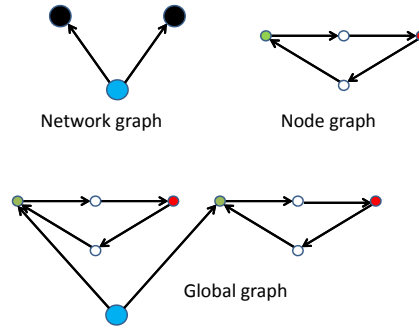


Figure 2: Example 2

Remark 3. *Although the node graphs and the network graph satisfy the controllability conditions of Theorem 1, the global system may be uncontrollable. This can be seen from the example in Figure 3, with again two identical node*

graphs. In this example the two conditions of Lin's Theorem are satisfied for the node graphs and for the network graph, but the system is not structurally controllable.

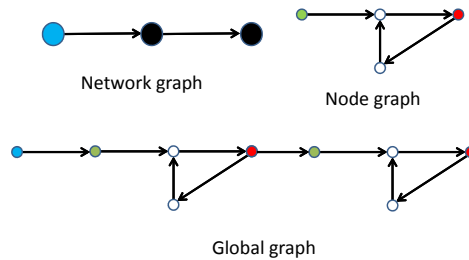


Figure 3: Example 3

From Remarks 2 and 3, it follows that the satisfaction of Lin's controllability conditions by the network, and by the node graphs, is neither a necessary, nor a sufficient condition, for the structural controllability of the global system.

5. Controllability of networks with identically structured nodes

In this Section, we particularize our study to the case where all nodes of the network have the same structure, and therefore the same graph. Such networks are sometimes called *homogeneous* in the literature [15]. Notice that this does not mean that the nodes are dynamically the same. In the case of truly identical nodes, some controllability problems may arise from symmetry considerations [30, 21]. For example, it is clear that if in the example of Figure 2, the two nodes have identical parameter values, the global system would not be controllable.

5.1. Main result for identically structured nodes

We can now state a simple necessary and sufficient condition for the controllability of a network with identically structured nodes.

Theorem 2. *Consider a network \mathcal{N} with N internal nodes, m control nodes with $N > m$, and its graph $G(\mathcal{N})$. Assume that all nodes are identically structured systems $\bar{\Sigma}_\Lambda$ with associated graph $G(\bar{\Sigma}_\Lambda)$. The global system is structurally controllable if and only if the following three conditions hold.*

1. *The graph $G(\mathcal{N})$ of the network is control connected.*

2. The node system $\bar{\Sigma}_\Lambda$ is structurally controllable.
3. And
 - a) either the vertices of the node graph $G(\bar{\Sigma}_\Lambda)$ can be covered by a disjoint set of cycles.
 - b) or the vertices of the node graph $G(\bar{\Sigma}_\Lambda)$ can be covered by a disjoint set of cycles and an in-out path, and the nodes of the network graph $G(\mathcal{N})$ can be covered by a disjoint set of stems and cycles.

Proof:

\Rightarrow Suppose that the global system is controllable. From Proposition 2, condition 1 must be satisfied, and from Proposition 3, condition 2 must be satisfied. Now suppose that condition 3-a is not satisfied, *i.e.* the global graph cannot be covered by a disjoint set of internal node cycles. On another hand, the global graph is covered by a disjoint set of stems and cycles. The assumption $N > m$ implies that the network graph cannot be covered by a set of stems of length one. Therefore a covering set of the global graph contains either cycles or stems of length at least two in $G(\mathcal{N})$, *i.e.* stems passing at least through one node. For such a cycle or stem passing through a node, the part of the stem, or of the cycle, which is contained in a node is an in-out path, and the other vertices of the node graph must be covered by internal node cycles, then the node graph $G(\bar{\Sigma}_\Lambda)$ can be covered by a disjoint set of cycles and an in-out path. Moreover, the disjoint set of stems and cycles covering the vertices of the global graph induces a disjoint set of stems and cycles covering the vertices of the network graph. Therefore, condition 3-b is satisfied.

Conversely, if condition condition 3-b is not satisfied, there cannot exist non purely node internal cycles, or stems of length at least two in the network graph, covering the vertices of the global graph. Therefore, the vertices of the global graph can be covered only by a disjoint set of node internal cycles and condition 3-a is satisfied.

\Leftarrow Suppose that conditions 1 and 2 are satisfied. The control connection of the global graph follows from Proposition 2. If condition 3-a is also satisfied, then the conditions of Lin's Theorem are satisfied and the global system is structurally controllable, see Proposition 4.

If, besides conditions 1 and 2, condition 3-b is satisfied, one can transform the disjoint set of stems and cycles covering the nodes of the network graph, into a disjoint set of stems and cycles covering the vertices of the global graph by using the in-out path to pass through each node and by adding the internal node cycles.

Then conditions 1, 2 and 3-a or 3-b, are sufficient for the structural controllability of the global graph. \blacksquare

The theorem simply says that, either the node graph can be covered by a disjoint set of node internal cycles, which insures the satisfaction of the second condition of Lin's Theorem for the global graph, or the covering of the global graph needs the use of the cycles and stems of the network graph. In the latter case, the node graph must "open the way to pass through it", by using an in-out path, while preserving the covering of other vertices by internal node cycles.

Remark 4. *The hypothesis of equality for the node graphs is mainly motivated by the simplicity of the formulation of Theorem 2. The result would remain valid if the vertices of all the node graphs can be covered either by a disjoint set of cycles, or by a disjoint set of cycles plus an in-out path. The extension to the general case is more complex and will be treated in the next Section.*

Remark 5. *The (reasonable) assumption $N > m$, was made for avoiding the situation where all the nodes can be covered by a stem of length one. In this case, the necessity of crossing the nodes which shows up in condition 3-b of Theorem 2 is no longer valid. This is illustrated by the example of Figure 4 where $N = m = 1$. It can then be seen that none of conditions 3-a and 3-b of Theorem 2 is satisfied, but the global system is controllable.*

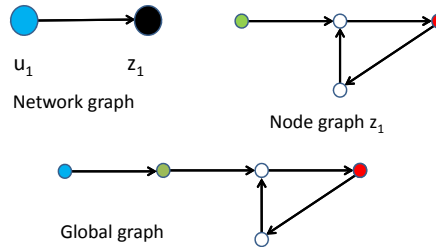


Figure 4: Example 4

Remark 6. *Although there is no particular hierarchy between conditions 3-a and 3-b of Theorem 2, it is clear that conditions 3-a is easier to check. Therefore, it is reasonable to check first conditions 3-a and, only if it is not satisfied, to check conditions 3-b. This is the sequence which will be developed in the next Subsection.*

5.2. Covering by cycles and an in-out path

In Theorem 2, condition 3-a, we are looking for a covering of the state vertices by a disjoint set of cycles. This can be tested by checking that there is a maximum matching of size n (called perfect matching in this case) in the bipartite graph $H(\bar{A}_\Lambda)$ associated with matrix \bar{A}_Λ , see Corollary 1. Following Remark 6, we will examine Condition 3-b, assuming that Condition 3-a is not satisfied. Condition 3-b of Theorem 2, implies to find a covering by a disjoint set of cycles and an in-out path, which is more restrictive than finding a covering set of disjoint cycles. Moreover, there is no direct algorithmic way to check this condition. Indeed, a simple trick allows to check the condition

using a maximum matching algorithm. Consider the matrix \bar{A}_Λ together with its graph $G(\bar{A}_\Lambda)$. Call x_i the in-vertex and x_j the out-vertex in $G(\bar{A}_\Lambda)$, if the edge (x_j, x_i) does not exist in $G(\bar{A}_\Lambda)$, let us transform the graph by adding an edge (x_j, x_i) to the graph $G(\bar{A}_\Lambda)$. This corresponds to adding a parameter in position (i, j) of the structured matrix \bar{A}_Λ transforming it in $\bar{A}_{\Lambda e}$, then we will call this modified graph $G(\bar{A}_{\Lambda e})$. This simple transformation allows to translate the covering by cycles and an in-out path into a standard problem of covering by cycles, which in turn can be checked by Corollary 1. This technique was frequently used in Reinschke's book [24].

Lemma 1. *Assume that the graph $G(\bar{A}_\Lambda)$ cannot be covered by a disjoint set of cycles. Then, the graph $G(\bar{A}_\Lambda)$ can be covered by a disjoint set of cycles and an in-out path, if and only if the graph $G(\bar{A}_{\Lambda e})$ can be covered by a disjoint set of cycles.*

Proof:

Notice first that if the edge (x_j, x_i) exists in $G(\bar{A}_\Lambda)$, and if the graph $G(\bar{A}_\Lambda)$ can be covered by a disjoint set of cycles and an in-out path, joining the in-out path and the edge (x_j, x_i) would create a new cycle in $G(\bar{A}_\Lambda)$, which contradicts our assumption.

Assume that $G(\bar{A}_\Lambda)$ cannot be covered by a disjoint set of cycles, but $G(\bar{A}_{\Lambda e})$ can be covered by a disjoint set of cycles. The disjoint set of cycles covering $G(\bar{A}_{\Lambda e})$ must contain a cycle including the out-in edge (x_j, x_i) . The other part of this cycle is an in-out path. Then, the graph $G(\bar{A}_\Lambda)$ can be covered by a disjoint set of cycles and an in-out path.

Conversely, if the graph $G(\bar{A}_\Lambda)$ can be covered by a disjoint set of cycles and an in-out path, adding the edge (x_j, x_i) provides with a disjoint set of cycles covering $G(\bar{A}_{\Lambda e})$. ■

6. Controllability of networks, the general case

In Section 5 we dealt with the situation where all nodes have the same structure. This assumption allowed to give a simple closed-form necessary and sufficient condition for controllability. We will see next that, using the same ideas and tools, the general case can be solved by the introduction of an auxiliary graph.

6.1. Node classification in the network graph

From the previous sections, we know that some properties of the node graphs are crucial in controllability issues. Recall first that, from Proposition 3, the controllability of each node system is a necessary condition for the controllability of the global graph. This means that each node graph must be covered by a disjoint set of cycles and possibly a stem. A node graph is said to be of type *CYC*, if all the state vertices can be covered by a disjoint set of node internal cycles, otherwise it is said to be of type \overline{CYC} . A node graph is said to be of type *IOP*, if the state vertices can be covered by a disjoint set of cycles and an

in-out path, otherwise it is said to be of type \overline{IOP} . Notice that for the IOP condition, the presence of an in-out path (possibly of length zero, when the in-vertex and the out-vertex are the same vertex) in the covering set is obligatory. The CYC and IOP nodes correspond to two particular situations for the second condition of Theorem 1. For CYC nodes, the covering of state vertices can be done without using a stem, while for IOP nodes the covering uses a particular stem which is an in-out path.

Since the properties CYC and IOP are not mutually exclusive, we can encounter all the possible combinations, *i.e.* $CYC.IOP$, $\overline{CYC}.IOP$, $CYC.\overline{IOP}$ and $\overline{CYC}.\overline{IOP}$. As an illustration, it can be seen that in Example 1, the node z_1 is $CYC.IOP$, the node z_2 is $\overline{CYC}.IOP$ and the node z_4 is $CYC.\overline{IOP}$.

From the results of the previous sections, one can make the following observation. When a node is IOP , it can be used in an object of the global graph (stem or cycle) which is also composed of edges of $G(\mathcal{N})$, the vertices which are not on the in-out path being covered by internal cycles. More precisely, a cycle of the network graph can be translated into a cycle of the global graph if and only if it crosses IOP nodes of $G(\mathcal{N})$. A stem of the network graph can be translated into a stem of the global graph if and only if it crosses IOP nodes of $G(\mathcal{N})$. Notice that the terminal node of the stem need not be IOP . The use of IOP nodes is illustrated on Figure 5 with a stem (u_1, z_1, z_2) and a cycle (z_3, z_4) of the network graph which are transformed in a stem and a cycle of the global graph.

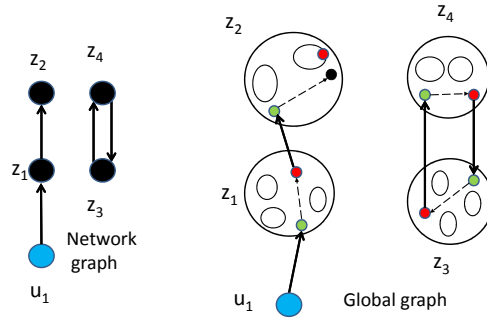


Figure 5: Illustration of the use of IOP nodes

6.2. The auxiliary network graph

Now, starting with a network \mathcal{N} , with associated graph $G(\mathcal{N})$ and using the properties of the node graphs, we will define a *network auxiliary graph* $G(\mathcal{N}a)$ as follows.

1. Initialize $G(\mathcal{N}a)$ as $G(\mathcal{N})$.

2. For each node z_i of type \overline{IOP} , delete from $G(\mathcal{N})$ the edges (z_i, z_j) for $j = 1 \dots, N$. This operation visualizes the fact that a cycle or a stem of the global graph cannot "cross the node z_i ".
3. For any node z_i of type CYC , add a self-loop to z_i in the graph $G(\mathcal{N})$. Such a self-loop is called a node self-loop, which should not be confused with a network self-loop which could pre-exist in $G(\mathcal{N})$.

Notice that for an \overline{IOP} node, the network self-loops are eliminated in step 2. If a node with a network self-loop is both CYC and IOP , this node would have two self-loops after steps 2 and 3, but, for simplicity, we will only keep one self-loop in $G(\mathcal{N}a)$.

6.3. The main result

With this definition of the auxiliary graph $G(\mathcal{N}a)$, we can now state the main result of the paper.

Theorem 3. *Consider a network \mathcal{N} with N internal nodes, m control nodes, and its graph $G(\mathcal{N})$; each internal node z_i of \mathcal{N} being defined by a structured system $\Sigma_{\Lambda,i}$, for $i = 1 \dots, N$.*

The global system Σ_{Λ} is structurally controllable if and only if the following three conditions hold.

1. *The graph $G(\mathcal{N})$ of the network is control connected.*
2. *The node system $\Sigma_{\Lambda,i}$ is structurally controllable, for $i = 1 \dots, N$.*
3. *The auxiliary graph $G(\mathcal{N}a)$ can be covered by a disjoint set of cycles and stems.*

Proof:

\Rightarrow Suppose that the global system is controllable, from Proposition 2, condition 1 must be satisfied, and from Proposition 3, condition 2 must be satisfied.

From the controllability of the global system, the graph $G(\Sigma_{\Lambda})$ can be covered by a disjoint set of cycles and stems. If a subset of the cycles covers all the vertices of the node graph $G(\Sigma_{\Lambda,i})$, this corresponds to the node self-loop of z_i in the auxiliary graph $G(\mathcal{N}a)$. If a cycle of $G(\Sigma_{\Lambda})$ passes through nodes $(z_{i_1}, \dots, z_{i_k})$, the corresponding nodes must be of type IOP . Then these nodes have conserved their outgoing edges in the second step of the construction of $G(\mathcal{N}a)$, which implies that the cycle of $G(\Sigma_{\Lambda})$ induces a cycle of $G(\mathcal{N}a)$.

Let us now consider a stem from the disjoint set of cycles and stems covering $G(\Sigma_{\Lambda})$. If this stem of $G(\Sigma_{\Lambda})$ visits nodes $(u_{\mu}, z_{j_1}, \dots, z_{j_{l-1}}, z_{j_l})$ of $G(\mathcal{N})$, the nodes $(z_{j_1}, \dots, z_{j_{l-1}})$ must be of type IOP . Then these nodes have conserved their outgoing edges in the second step of the construction of $G(\mathcal{N}a)$, which implies that the stem of $G(\Sigma_{\Lambda})$ induces a stem of $G(\mathcal{N}a)$. Therefore, the disjoint set of cycles and stems covering $G(\Sigma_{\Lambda})$ induces a disjoint set of cycles and stems covering $G(\mathcal{N}a)$.

\Leftarrow Suppose that conditions 1 and 2 are satisfied. The control connection of the global graph follows from Proposition 2.

Consider a disjoint set of cycles and stems covering $G(\mathcal{N}a)$, with a particular

stem $(u_\mu, z_{j_1}, \dots, z_{j_{i-1}}, z_{j_i})$. Each node of this stem can be replaced by the corresponding in-out path in the node graph to form a stem of the global graph $G(\Sigma_\Lambda)$. Notice that in the end node of the stem, z_{j_i} , we can use the stem of the corresponding node graph to complete the stem of $G(\Sigma_\Lambda)$. From the definition of an *IOP*, all the other vertices of the nodes in the stem are covered by cycles. The same type of argument works for cycles of $G(\mathcal{N}a)$ composed of more than one node or for network self-loops. If a node self-loop belongs to the disjoint set of cycles and stems covering $G(\mathcal{N}a)$, from the definition of a node self-loop the vertices of this node are covered by a disjoint set of internal cycles. Therefore, from the disjoint set of cycles and stems covering $G(\mathcal{N}a)$, one can build a disjoint set of cycles and stems covering the global graph $G(\Sigma_\Lambda)$. The structural controllability of Σ_Λ follows. ■

Example 1 continued If we consider again Example 1, we can classify the different nodes, and then get the auxiliary graph. Nodes z_2 and z_3 are controllable but not *IOP*, therefore the outgoing edges of these two nodes are deleted from $G(\mathcal{N})$. Nodes z_4 and z_5 are *CYC* but not *IOP*, therefore the outgoing edges of these two nodes are deleted from $G(\mathcal{N})$ and node self-loops are added to z_4 and z_5 in $G(\mathcal{N}a)$. Node z_1 is *CYC*, therefore a node self-loop is added to z_1 in $G(\mathcal{N}a)$. The resulting auxiliary graph $G(\mathcal{N}a)$ is given in Figure 6. It is easily seen that $G(\mathcal{N}a)$ can be covered by a disjoint set of cycles and stems (stems (u_1, z_1, z_2) and (u_2, z_3) , together with the self-cycles on z_4 and z_5) therefore the global system is structurally controllable. It is important to note that the auxiliary graph $G(\mathcal{N}a)$ need not be control-connected.

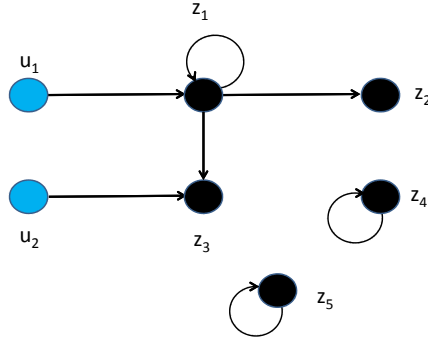


Figure 6: Auxiliary graph for Example 1

6.4. Finding *CYC* and *IOP* nodes

Here, the situation is less simple than in the previous Section. In Section 5, it is clear that, when a node graph is *CYC*, there was no need to check for the *IOP* property. Now, both properties may be useful for controllability of the

global system, and must be tested for each node. For the property *CYC*, this needs checking that there is a maximum matching of size n_i in the bipartite graph $H(A_{\Lambda,i})$ associated with matrix $A_{\Lambda,i}$ of node system $\Sigma_{\Lambda,i}$.

For testing the *IOP* property, we formulate it as a maximum matching with minimum cost problem [31]. The idea is to force, if possible, the use of an in-out path in a maximum matching. Consider the matrix $A_{\Lambda,i}$ with its graph $G(A_{\Lambda,i})$ and bipartite graph $H(A_{\Lambda,i})$. Let x_k and x_l be the in-state and out-state respectively. If there is no edge (x_l, x_k) in the graph $G(A_{\Lambda,i})$, add a dummy edge (x_l, x_k) , and do the same operation in the bipartite graph $H(A_{\Lambda,i})$. Give to all edges of $H(A_{\Lambda,i})$ a cost one, except for the edge (x_l, x_k) (real or dummy) which has cost zero. Call this new weighted bipartite graph $H(A_{\Lambda,i}^w)$ and apply on it a maximum matching minimum cost algorithm. The characterization of the *IOP* nodes then follows.

Lemma 2. *The node graph $G(\Sigma_{\Lambda,i})$ has the property *IOP* if and only if there exists a size n matching with cost $(n - 1)$ in the graph $H(A_{\Lambda,i}^w)$.*

Proof:

If the node graph $G(\Sigma_{\Lambda,i})$ has the property *IOP*, completing the corresponding in-out path by the edge (x_l, x_k) , provides with a disjoint set of cycles covering all state vertices of $G(\Sigma_{\Lambda,i})$, *i.e.* with a matching of size n in $H(A_{\Lambda,i}^w)$. Moreover this matching has a minimum cost $(n - 1)$.

Conversely, if there exists a size n matching with cost $(n - 1)$ in the graph $H(A_{\Lambda,i}^w)$, this set must contain (x_l, x_k) which is the only edge with cost zero. Therefore the vertices of $G(A_{\Lambda,i})$ can be covered by a disjoint set of cycles and an in-out path, then $\Sigma_{\Lambda,i}$ is *IOP*. ■

6.5. Minimal Controllability Problems

Starting from an autonomous system, and wondering how many (ideally few) external inputs, and how (ideally not too intrusively) these inputs should act on the system to make it controllable, is a question that has rised a tremendous number of works in the recent years. Declining these questions for graph-based systems has stemmed several variants which are very stimulating combinatorial problems, and which are known as *Minimal Controllability Problems* (MCP) [4, 5, 32, 33, 34, 35, 15, 36, 37].

Let us now summarize how the MCP can apply to our framework.

1. Check first that each node graph is controllable, and check if it is of *CYC* or *IOP* type.
2. Construct the auxiliary graph $G(\mathcal{N}a)$.
3. On the auxiliary graph $G(\mathcal{N}a)$, and its associated bipartite graph $H(\mathcal{N}a)$, the Minimum Inputs Theorem of [4] can then be applied directly. The technique of [34, 35], which is devoted to the case where an input can only impact one state vertex, can be applied with a slight subtlety. The maximum matching technique has to be applied on an extended bipartite graph composed of $H(\mathcal{N}a)$ completed with dummy nodes representing the critical connection components of $G(\mathcal{N})$ (not of $G(\mathcal{N}a)$), and the corresponding edges.

In summary, solving the MCP in our framework induces some preliminary tests on the node graphs (controllability and determination of their *CYC* and *IOP* properties), but otherwise, it essentially amounts to solve the MCP on an auxiliary graph which has the same size as the network graph.

Remark 7. *In practice, although there may exist vertices with no self-loop in a node graph, it will very often happen that the vertices can be covered by a disjoint set of cycles. This is the case if the nodes have asymptotically stable dynamics, which will happen for a lot of systems in biology or other fields of application. Asymptotic stability implies that each $A_{\Lambda,i}$ has no zero structural eigenvalue, i.e. that $A_{\Lambda,i}$ has generic rank n_i and therefore that $G(A_{\Lambda,i})$ can be covered by a disjoint set of cycles. In such a situation, all the nodes are *CYC* and, as argued in [6], Minimum Input Theorem has a solution with a unique input.*

7. On the complexity aspects

In this Section we study the complexity of our approach. In particular, we examine to which extent it is preferable to the brute force technique which consists in directly checking the controllability of the global graph.

Let us first recall that the controllability of a structured system implies checking the two properties of Theorem 1. The connection property can be checked using a depth first algorithm whose complexity is in $O(n^2)$, where n is the number of states vertices in the graph $G(\Sigma_\Lambda)$. Notice that this complexity corresponds to the worst case and that the algorithm is much more efficient if $G(\Sigma_\Lambda)$ is sparse. The second condition, as mentioned in Subsection 3.2, can be checked by looking for a maximum matching in the associated bipartite graph $H(\Sigma_\Lambda)$. Assuming that the number of controls is less than the number of states, *i.e.* $m < n$, the complexity of the maximum matching is in $O(n^{2.5})$. Notice that the maximum matching problem is still an active field of research and that the previous complexity may be improved by taking into account some features of the graph, see [34, 37] for more discussions on this point. In summary, the structural controllability can be checked with complexity $O(n^{2.5})$, where n is the number of states vertices in the graph $G(\Sigma_\Lambda)$.

Let us now examine our situation, where the system is a network of size N defined by its graph $G(\mathcal{N})$, each node z_i of $G(\mathcal{N})$ being defined by a structured system with its own graph $G(\Sigma_{\Lambda,i})$. We check the controllability of the global graph using our approach, *i.e.* by first studying each node separately and then by dealing with the auxiliary graph as in Theorem 3. To allow simple complexity comparisons, assume that each node graph has the same number of state vertices \bar{n} (but not necessarily the same graph). For control connection of the global graph one can use Proposition 2. The connection of each node can be checked in $O(\bar{n}^2)$ and the connection of the network graph $G(\mathcal{N})$ can be checked in $O(N^2)$. Therefore the global connection can be checked in $O(N\bar{n}^2 + N^2)$. Assuming that the size of a node graph is much lower than the size of the network, it follows that checking the control connection is made with complexity $O(N^2)$.

The control connection of the global graph (whose dimension is $N\bar{n}$) by a direct method would have complexity $O(N^2\bar{n}^2)$.

For checking the second structural controllability condition on the global graph, we have first to study each node graph. This implies checking the structural controllability of the node graph and the *CYC* and *IOP* properties. As seen before, the structural controllability of the node graph and the *CYC* property can be checked with complexity $O(\bar{n}^{2.5})$, since they only involve a maximum matching algorithm. The *IOP* property, as seen in Lemma 2, needs an algorithm finding a maximum matching with a minimum cost. This type of algorithm has complexity $O(\bar{n}^3)$ [38]. The complete study of a node has then complexity $O(\bar{n}^3)$. From Theorem 3, the second structural controllability condition of the global graph implies the study of each node graph and checking the second structural controllability condition on the network auxiliary graph $G(\mathcal{N}_a)$. This globally implies a complexity in $O(N\bar{n}^3 + N^{2.5})$. With our assumption that the size of node graph is much lower than the size of the network, it follows that the second structural controllability condition on the global graph can be checked with complexity $O(N^{2.5})$. The second structural controllability condition on the global graph (whose dimension is $N\bar{n}$) by a direct method would have complexity $O(N^{2.5}\bar{n}^{2.5})$.

In summary, we assumed here that all nodes have \bar{n} vertices, and that \bar{n} is much smaller than the number N of nodes in the network graph $G(\mathcal{N})$. In this case, checking the structural controllability by the previous method is roughly $\bar{n}^{2.5}$ times less costly than checking the structural controllability in a classical way on the global graph. As an example, for a large N and $\bar{n} = 10$, the computational load would be divided by more than 300. Even if the nodes have not the same number of vertices, our approach remains very efficient as long as \bar{n} is much smaller than the number N . Indeed, with our approach the controllability problem for a size N network has just the complexity of a maximum matching problem for a size N bipartite graph.

8. Conclusion

In this paper, we addressed the controllability of a network whose nodes are composed of a single-input, single-output structured system. We have then to deal with a global graph which is the composition of the network graph and of the graph of each node. We gave simple necessary and sufficient controllability conditions in terms of these different graphs. Assuming that the node graphs have a modest dimension, it appears that the complexity of the associated computations is only related to the size of the network. Also, the Minimal Controllability Problems in this framework, are not significantly harder than in the usual case. This work may be a starting point for the study of more complex node systems or more sophisticated connection rules.

Acknowledgment

The author wishes to thank Jacob van der Woude and Taha Boukhobza for several useful remarks and suggestions on a first version of this paper. He also

wishes to thank Jean-Michel Dion for a long collaboration and a lot of discussions on the ideas and concepts which lead to this paper.

9. References

- [1] S. Strogatz, “Exploring complex networks,” *Nature*, vol. 410, pp. 268–276, 2001.
- [2] F. Sorrentino, M. M. di Bernardo, F. Garofalo, and G. Chen, “Controllability of complex networks via pinning,” *Phy. Rev. E*, vol. 75, no. 046103, 2007.
- [3] A. Lombardi and M. Hornquist, “Controllability analysis of networks,” *Phy. Rev. E*, vol. 75, no. 056110, 2007.
- [4] Y. Y. Liu, J. J. Slotine, and A. L. Barabasi, “Controllability of complex networks,” *Nature*, vol. 473, pp. 167–173, 2011.
- [5] J. Ruths and D. Ruths, “Control profiles of complex networks,” *Science*, vol. 343, pp. 1373–1375, 2014.
- [6] N. J. Cowan, E. J. Chastain, D. A. Vilhana, J. S. Freudenberg, and C. T. Bergstrom, “Nodal degree, not degree distributions, determine the structural controllability of complex networks,” *PLOS ONE*, vol. DOI: 10.1371, 2012.
- [7] T. Nepusz and T. Vicsek, “Controlling edge dynamics in complex networks,” *Nature Physics*, no. 8, pp. 568–573, 2012.
- [8] Y. Wang, J. Xiang, Y. Li, and M. Chen, “Controllability of dynamic-edge multi-agent systems,” *IEEE Trans. Cont. of Net. Syst.*, To appear 2017.
- [9] E. Davison and S. Wang, “New results on the controllability and observability of general composite systems,” *IEEE Trans. Automat. Control*, vol. 20, no. 1, pp. 123–128, 1976.
- [10] E. Davison, “Connectability and structural controllability of composite systems,” *Automatica*, vol. 13, no. 2, pp. 109–123, 1977.
- [11] C. Rech and R. Perret, “About structural controllability of interconnected dynamic systems,” *Automatica*, vol. 27, no. 5, pp. 201–208, 1991.
- [12] L. Blackhall and D. Hill, “On the structural controllability of networks of linear systems,” in *2nd IFAC Workshop on Distributed Estimation and Control in Networked Systems, NecSys*, Annecy, France, 2010.
- [13] T. Zhou, “On the controllability and observability of networked dynamic systems,” *Automatica*, vol. 52, pp. 63–75, 2015.
- [14] L. Wang, G. Chen, X. Wang, and W. Tang, “Controllability of networked MIMO systems,” *Automatica*, vol. 69, pp. 405–409, 2016.

- [15] J. Carvalho, S. Pequito, P. Aguiar, S. Kar, and K. Johansson, “Composability and controllability of linear time-invariant systems: Distributed verification,” *Automatica*, vol. 78, no. 4, pp. 123–134, 2017.
- [16] A. Chapman, M. Nabi-Abdolyousefi, and M. Mesbahi, “Controllability and observability of networks of networks via Cartesian products,” *IEEE Trans. Automat. Control*, vol. 59, no. 10, pp. 2668–2679, 2014.
- [17] C. T. Lin, “Structural controllability,” *IEEE Trans. Automat. Control*, vol. 19, no. 3, pp. 201–208, 1974.
- [18] K. Murota, *Systems Analysis by Graphs and Matroids*, ser. Algorithms and Combinatorics. Springer-Verlag New-York, Inc., 1987, vol. 3.
- [19] J. M. Dion, C. Commault, and J. van der Woude, “Generic properties and control of linear structured systems: a survey,” *Automatica*, vol. 39, no. 7, pp. 1125–1144, 2003.
- [20] Z. Yuan, C. Zhao, Z. Di, W.-X. Wang, and Y.-C. Lai, “Exact controllability of complex networks,” *Nature Communications*, vol. 4:2447, no. doi:10.1038/ncomms3447, 2013.
- [21] L. Wang, X. Wang, and G. Chen, “Controllability of networked higher-dimensional systems with one-dimensional communication,” *Phil. Trans. R. Soc. A*, vol. 375:20160215, pp. 2–10, 2017.
- [22] K. Li, Y. Xi, and Z. Zhang, “G-cactus and new results on structural controllability of composite systems,” *International Journal of Systems Science*, vol. 27, no. 12, pp. 1313–1326, 1996.
- [23] R. Shields and J. Pearson, “Structural controllability of multi-input linear systems,” *IEEE Trans. Automat. Control*, vol. AC-21, pp. 203–212, 1976.
- [24] K. Reinschke, *Multivariable control: a graph-theoretic approach*. Springer Verlag, 1988.
- [25] R. Tarjan, “Depth-first search and linear graph algorithms,” *SIAM J. Computing*, vol. 1, no. 2, pp. 146–160, 1972.
- [26] J. Hopcroft and R. Karp, “An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs,” *SIAM J. on Comp.*, vol. 2, pp. 225–231, 1973.
- [27] A. Madry, “Navigating central path with electrical flows: from flows to matchings, and back,” in *IEEE Symposium on Foundations of Computer Science (FOCS)*, 2013, pp. 253–262.
- [28] M. Mucha and P. Sankowski, “Maximum matchings via gaussian elimination,” in *IEEE Symposium on Foundations of Computer Science (FOCS)*, 2004, pp. 248–255.

- [29] C. Commault, J. M. Dion, and J. van der Woude, “Characterization of generic properties of linear structured systems for efficient computations,” *Kybernetika*, vol. 38, no. 5, pp. 503–520, 2002.
- [30] A. Chapman and M. Mesbahi, “On symmetry and controllability of multi-agent systems,” in *IEEE CDC Conference*, Los Angeles, 2014, pp. 625–630.
- [31] H. W. Kuhn, “The hungarian method for the assignment problem,” *Naval Research Logistic Quarterly*, vol. 2, pp. 83–97, 1955.
- [32] C. Commault and J. M. Dion, “Input addition and leader selection for the controllability of graph-based systems,” *Automatica*, vol. 49, no. 11, pp. 3322–3328, 2013.
- [33] —, “The single-input minimal controllability problem for structured systems,” *Syst. Cont. Lett.*, vol. 80, pp. 50–55, 2015.
- [34] S. Assadi, S. Khanna, Y. Li, and V. M. Preciado, “Complexity of the minimum input selection problem for structural controllability,” in *5th IFAC Workshop on Distributed Estimation and Control in Networked Systems, NecSys*, Philadelphia, 2015.
- [35] S. Pequito, S. Kar, and P. Aguiar, “A framework for structural input/output and control configuration selection in large-scale systems,” *IEEE Trans. Automat. Control*, vol. 61, no. 2, pp. 303–318, 2016.
- [36] A. Olshevsky, “Minimal controllability problems,” *IEEE Trans. on Control of Network Systems*, vol. 1, no. 3, pp. 249–258, 2014.
- [37] —, “Minimum input selection for structural controllability,” in *American Control Conference (ACC)*, 2015, pp. 2218–2223.
- [38] J. Edmonds and R. Karp, “Theoretical improvements in algorithmic efficiency for network flow problems,” *Journal of the ACM*, vol. 19, no. 2, pp. 248–264, 1972.