Strong Structural Input and State Observability of Linear Time-Invariant Systems: Graphical Conditions and Algorithms

Federica Garin, Sebin Gracy, Alain Kibangou

Strong Structural Input and State Observability of Linear Time-Invariant Systems: Graphical Conditions and Algorithms

Federica Garin[a,*], Sebin Gracy[b], Alain Y. Kibangou[a]

[a] *Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, GIPSA-lab, Grenoble, France*
[b] *Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden*

Abstract

The paper studies input and state observability (ISO) of discrete-time linear time-invariant network systems whose dynamics are affected by unknown inputs. More precisely, we aim at reconstructing the initial state and the sequence of unknown inputs from the system outputs, and we will use the term ISO when the input reconstruction is possible with delay one, namely the inputs up to time $k-1$ and the states up to time k can be obtained from the outputs up to time k, while the term unconstrained ISO will refer to the case where there is some arbitrary delay in the input reconstruction. We focus on the problem of s-structural ISO (resp. s-structural unconstrained ISO) wherein the objective is to find conditions such that for all system matrices that carry the same network structure, the resulting system is ISO (resp. unconstrained ISO). We provide first a graphical characterization for s-structural unconstrained ISO, and subsequently, sufficient conditions and necessary conditions for s-structural ISO. For the latter, under the assumption of zero feedthrough, these conditions coincide and characterise ISO. The conditions presented are in terms of existence of suitable uniquely restricted matchings in bipartite graphs associated with the structured system. In order to test these conditions, we present polynomial-time algorithms. Finally, we discuss an equivalent reformulation of the main conditions in terms of coloring algorithms as in the literature of zero forcing sets.

Keywords: Linear time-invariant systems, Network systems, Input and state observability, Structured systems, Strong structural observability, Uniquely restricted matchings, Constrained matchings, Zero forcing sets

1. Introduction

Modern life relies on critical infrastructure systems that provide essential services. These systems are more and more networked. It is vital to monitor efficiently such systems in order to promptly respond to failures and attacks they could face. For this purpose it is not only necessary to estimate the system states from sensor measurements but also reconstruct the possible unknown inputs affecting the system; these inputs representing failures or intentional attacks.

In a systems-theory context, we are interested in studying *Input and State Observability* (ISO), namely to understand whether it is possible to reconstruct both the initial state and the unknown input sequence, from the output sequence. Throughout this paper, the term *ISO* will be used for the case where the input reconstruction only has delay one (with the knowledge of outputs up to current time step k, we can reconstruct the input up to time $k-1$ and the state up to time k), while the term *unconstrained ISO* will be used for the case that some arbitrary delay is allowed in the input reconstruction.

Algebraic characterizations of ISO, in a similar vein as Kalman or PBH tests for controllability, are classical [1, 2, 3]. However these conditions, based on matrix rank computations, suffer from the following drawbacks: checking these conditions entails exact knowledge of the coefficients of the system matrices—a luxury often not available in network systems—and moreover it is non-trivial for large networks, due to ill-conditioning and complexity issues. Therefore, a rich trend of research looks for results based on structured systems theory. A structured system is a linear system whose system matrices have positions that are a priori fixed to zero, while the positions that are not a priori fixed to zero are *free parameters.* Under such a setting, there are two kinds of results, generally formulated as graphical conditions: i) *structural* results, that are true for almost all choices of free parameters, and ii) *strong structural (s-structural)* results, that are true for every choice of non-zero parameters. The understanding is as follows: structural results ensure that some property is true with probability one, if the free parameters are chosen at random from any continuous distribution. S-structural results, on the other hand, require that the property be guaranteed true, as far as all parameters are non-zero. The

*Corresponding author
Email addresses: `federica.garin@inria.fr` (Federica Garin), `gracy@kth.se` (Sebin Gracy), `alain.kibangou@univ-grenoble-alpes.fr` (Alain Y. Kibangou)

present paper is focused on s-structural results.

S-structural controllability (and, by duality, also observability) has been studied since the late 70s, starting with the seminal paper [4], where the single-input case was addressed. This was extended for the multi-input case in [5]. The results in [4, 5] are graph-theoretic conditions without easy algorithms to test them. More recently, building on the works [6, 7], a characterization in terms of existence of uniquely restricted matchings of appropriate size on suitably-defined bipartite graphs has been provided in [8], together with a polynomial-time algorithm to test such condition. This approach has been used in further works such as [9, 10]. Yet another approach towards studying s-structural controllability is that of zero forcing sets [11, 12, 13, 14]. The two approaches, with uniquely restricted matchings and with zero forcing sets, give characterizations which are equivalent, although very differently phrased, as first shown in [12]. In this paper, we will mostly focus on uniquely restricted matchings, but we will also show how to rephrase our main results with zero forcing sets.

The literature concerning s-structural ISO is, comparatively, rather thin. The results in [15] concern both linear time-invariant (LTI) and linear time-varying (LTV) systems with fixed graphs, but are restricted to systems that satisfy the following assumptions: each unknown input acts on a single state, no two inputs act on a same state, dedicated sensors are available (i.e., each measurement concerns a single state), and there is no direct feedthrough from input to output. The results in [16] do not require the aforementioned assumptions, but concern LTV systems only (either with time-varying or with time-invariant graph), while in this paper we are interested in LTI systems. A preliminary result about LTI systems can be inferred from the results in [16]: by considering the case of a time-invariant graph, the sufficient condition in [16, Thm. 2] for s-structural ISO of an LTV system is a sufficient condition also for s-structural ISO of the LTI system with the same graph. Indeed, this condition ensures that the system is ISO for all possible non-zero parameters, irrespective of how they vary in time, and hence in particular for all time-invariant ones. However, this condition involves the so-called dynamic bipartite graph, which describes the system evolution over a time interval as long as the number n of states and hence has a number of vertices that grows quadratically with n. The use of the dynamic bipartite graph is natural for a time-varying problem, while for the time-invariant case it would be desirable to have results involving a smaller graph, with a number of vertices growing only linearly with n and hence comparable with the size of the problem description. Moreover, the above-discussed condition is only sufficient, and there is no simple way to derive a necessary condition for the LTI case from the LTV one.

Paper Contributions

The present paper gives sufficient conditions and necessary conditions for s-structural ISO of LTI systems, without the restrictive assumptions considered in [15], and involving smaller graphs than the dynamic bipartite graph considered in [16]. Our conditions are expressed in terms of existence of suitable uniquely restricted matchings on bipartite graphs associated with the structured system. We also present an algorithm to find such matchings, whose complexity is linear in the total number of edges and vertices of the bipartite graph, and we discuss an alternative formulation involving color change rules as in the literature on zero forcing sets.

Our main results are the following. First, we provide a characterization of s-structural unconstrained ISO (Theorem 1). Then, for s-structural ISO: Each of the conditions in Theorem 1 and in Theorem 2 is necessary, while the conditions in Theorem 1 and Theorem 3, together, are sufficient. For the particular case of systems with no direct feedthrough, by combining the conditions in Theorem 1 and Theorem 4, we obtain a full characterization of s-structural ISO. For general feedthrough, instead, there is a gap between the necessary and sufficient conditions, as we highlight with relevant examples.

A preliminary version of this paper has been presented as [17]. The current paper improves upon the main results in [17] by providing a new sufficient condition for s-structural delay-1 left invertibility (Theorem 3), and showing that the old condition (presented here as Prop. 1) implies the new one in a non-trivial way. This paper is enriched with examples that not only illustrate the application of our main results, but also prove some relevant facts about s-structural delay-1 left invertibility: the two above-mentioned sufficient conditions are not equivalent to each other, nor they are necessary, and the necessary condition given in Theorem 2 is not sufficient. Furthermore, the sections on algorithms and their complexity and the results involving coloring rules and zero forcing sets are novel.

Paper Outline

The rest of this paper is organized as follows. We formally state the problem of interest in Section 2. The main results are stated in Section 3 and an algorithmic reformulation of the same is given in Section 4, together with a further reformulation involving coloring rules and zero forcing sets. Sect. 6 illustrates with examples the gaps between the sufficient and the necessary conditions for s-structural delay-1 observability. Sect. 5 introduces the main tools for the proofs (also useful in the discussion of the examples in Sect. 6), and the details of the proofs are given in Sections 7, 8 and 9. Finally, in Sect. 10 we summarize our results and discuss some possible directions for future research. The appendix discusses algorithmic complexity.

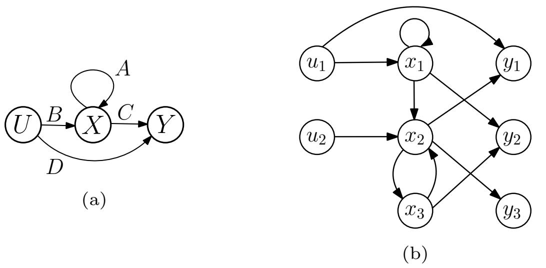


Figure 1: Directed graph $\mathcal{G}$. (a) Sketch of its construction; (b) Example 1.

2. Problem formulation

Consider a discrete-time LTI network system, whose dynamics are given by:

$$\begin{cases} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k \end{cases} \tag{1}$$

with state vector $x_k \in \mathbb{R}^n$, unknown input vector $u_k \in \mathbb{R}^p$ and output vector $y_k \in \mathbb{R}^m$. Matrices A B, C and D have some positions that are a priori fixed to zero, and all other positions occupied by free parameters (that is, distinct real-valued parameters, which can be chosen arbitrarily). The position of zeros represents the interactions that cannot happen in this network system, while parameters represent the intensity of existing interactions. Such a linear system with given zero positions and with free parameters is called a *structured system*. A structured system is usually described with a directed graph $\mathcal{G}$, with vertex set $U \cup X \cup Y$, where $U = \{u_1, \ldots, u_p\}$ are the input vertices, $X = \{x_1, \ldots, x_n\}$ are the state vertices, and $Y = \{y_1, \ldots, y_m\}$ are the output vertices. Edges of $\mathcal{G}$ correspond to the non-zero entries of matrices A, B, C, D: a non-zero entry a_{ij} of matrix A corresponds to an edge (x_j, x_i), representing the influence of state $x_j(k)$ on the state $x_i(k+1)$, a non-zero entry b_{ij} of matrix B corresponds to an edge (u_j, x_i), a non-zero entry c_{ij} of matrix C corresponds to an edge (x_j, y_i), and a non-zero entry d_{ij} of matrix D corresponds to an edge (u_j, y_i). Figure 1a gives a pictorial reminder of this construction, while an example is the following.

Example 1. *Consider the system* (1)*, with matrices*

$$A = \begin{bmatrix} a_{11} & 0 & 0 \\ a_{21} & 0 & a_{23} \\ 0 & a_{32} & 0 \end{bmatrix}, \quad B = \begin{bmatrix} b_{11} & 0 \\ 0 & b_{22} \\ 0 & 0 \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & c_{12} & 0 \\ c_{21} & 0 & c_{23} \\ 0 & c_{32} & 0 \end{bmatrix}, \quad D = \begin{bmatrix} d_{11} & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

where all non-zero entries are independent real-valued parameters. This is a structured system, whose directed graph $\mathcal{G}$ is in Figure 1b.

This paper studies the system properties for reconstruction of both unknown inputs and states from outputs.

Definition 1. *System* (1) *is* unconstrained ISO *if there exists some integer $\ell \geq 0$ such that the initial condition x_0 and the unknown inputs $u_0, u_1, \ldots, u_{n-\ell}$ can be uniquely determined from the outputs $y_0, y_1, \ldots, y_n$.* ■

In case the input reconstruction delay ℓ is constrained to be equal to one, we will say that the system is ISO:

Definition 2. *System* (1) *is* ISO *if the initial condition x_0 and the unknown inputs $u_0, u_1, \ldots, u_{n-1}$ can be uniquely determined from the measured outputs $y_0, y_1, \ldots, y_n$.* ■

Notice that ISO is equivalent to unconstrained ISO together with the delay-1 left invertibility property, which is defined as follows:

Definition 3. *System* (1) *is* delay-1 left invertible *if the input u_0 can be uniquely determined from the initial state x_0 and the outputs y_0 and y_1.* ■

The goal of this paper is to understand whether a structured system is (unconstrained) ISO for all possible non-zero choices of its free parameters. In such a case, the system will be said to be s-structurally *(unconstrained) ISO.*

3. Graphical conditions for s-structural ISO

In this section, we give the statement of our main results: graphical conditions which fully characterize s-structural unconstrained ISO, some sufficient and some necessary conditions for s-structural delay-1 left invertibility, and the characterization of s-structural delay-1 left invertibility in the case where $D = 0$. Recalling that ISO is equivalent to unconstrained ISO together with delay-1 left invertibility, these conditions together provide conditions for s-structural ISO.

3.1. Uniquely restricted matchings

Our results rely on the notion of uniquely restricted matching, which we recall in this section together with a few related definitions.

Given a graph, a *matching* is a subset of edges such that no two edges share a common vertex. The size of a matching is its number of edges; if a matching has maximum size among all the matchings in the same graph, then it is a *maximum matching*. We will say that a matching *covers* those vertices that are an end-point of an edge in the matching, and we will say that a vertex is *matched* if it is covered by the matching, and *unmatched* otherwise. Similarly we will say that an edge is matched if it belongs to the matching and is unmatched otherwise.

Uniquely restricted matchings (also known as *constrained* matchings) are defined as follows.

Definition 4 ([18, Definition 2.4]). *Given a graph with edge set $\mathcal{E}$, a matching $\mathcal{M} \subseteq \mathcal{E}$ is* uniquely restricted *if there exists no other matching $\mathcal{M}' \subseteq \mathcal{E}$ covering exactly the same set of vertices as $\mathcal{M}$.* ■

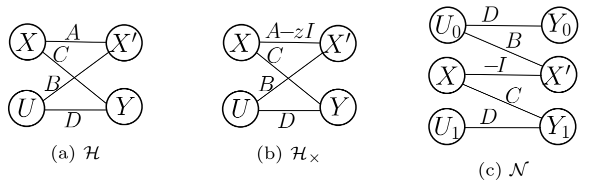


Figure 2: Sketch of the construction of $\mathcal{H}$, $\mathcal{H}_\times$, and $\mathcal{N}$. From $\mathcal{N}$, $\tilde{\mathcal{N}}$ is obtained by removing U_1 and its incident edges, and $\mathcal{N}_0$ by removing U_1, Y_0 and their incident edges.

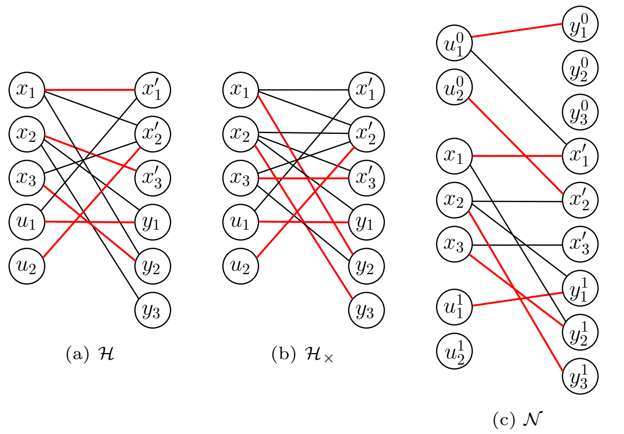


Figure 3: Bipartite graphs $\mathcal{H}$, $\mathcal{H}_\times$, $\mathcal{N}$ for Example 1. In $\mathcal{H}_\times$, $\mathcal{E}_{\text{loop}} = \{(x_1, x_1')\}$ and $\mathcal{E}_{\text{new}} = \{(x_2, x_2'), (x_3, x_3')\}$. Red edges represent the uniquely restricted matchings $\mathcal{M}$, $\mathcal{M}_\times$ and $\mathcal{L}$ described in Sect. 3.

As shown in [19, Thm. 2.1], a matching $\mathcal{M}$ is uniquely restricted if and only if the graph does not contain any alternating cycle with respect to $\mathcal{M}$, i.e., any cycle of even length with edges alternating between matched and unmatched edges.

3.2. Characterization of s-structural unconstrained ISO

In this section, we present a graphical characterization of s-structural unconstrained ISO. The graphs involved in its statement are not the directed graph $\mathcal{G}$ introduced in Sect. 2, but two bipartite graphs defined as follows.

The bipartite graph $\mathcal{H}$ has left vertex set $X \cup U$ and right vertex set $X' \cup Y$, where U, X, and Y are the same input, state, and output vertex sets used in the definition of the directed graph $\mathcal{G}$, while $X' = \{x_1', \ldots, x_n'\}$ is a copy of the state vertex set X. Edges of $\mathcal{H}$ are in one-to-one correspondence with non-zero entries of matrices A, B, C, D, as follows: $a_{ij} \neq 0$ corresponds to an edge (x_j, x_i'), $b_{ij} \neq 0$ corresponds to an edge (u_j, x_i'), $c_{ij} \neq 0$ corresponds to an edge (x_j, y_i), and $d_{ij} \neq 0$ corresponds to an edge (u_j, y_i). Notice that edges of $\mathcal{H}$ are in one-to-one correspondence with edges of $\mathcal{G}$.

The bipartite graph $\mathcal{H}_\times$ is obtained from $\mathcal{H}$ by extending its edge set as follows. We will use the word 'self-loops' to indicate edges (x_i, x_i'), since such edges correspond to actual self-loops (x_i, x_i) in $\mathcal{G}$. We define $\mathcal{E}_{\text{loop}}$ as the set of 'self-loops' that are edges of $\mathcal{H}$, and $\mathcal{E}_{\text{new}}$ as the set of 'self-loops' that are not edges of $\mathcal{H}$. We obtain $\mathcal{H}_\times$ from $\mathcal{H}$, by adding all edges of $\mathcal{E}_{\text{new}}$.

A sketch of the construction of $\mathcal{H}$ and $\mathcal{H}_\times$ is given in Figures 2a and 2b, while Figures 3a and 3b show $\mathcal{H}$ and $\mathcal{H}_\times$ for Example 1.

With these definitions in place, we have the following characterization of s-structural unconstrained ISO.

Theorem 1. *The structured system* (1) *is s-structurally unconstrained ISO if and only if the following conditions are satisfied:*

i. *in $\mathcal{H}$ there exists a uniquely restricted matching $\mathcal{M}$ of size $n + p$, and*
ii. *in $\mathcal{H}_\times$ there exists a uniquely restricted matching $\mathcal{M}_\times$ of size $n + p$, such that $\mathcal{M}_\times \cap \mathcal{E}_{\text{loop}} = \emptyset$,*

where $n = |X|$ and $p = |U|$. ■

The conditions in Theorem 1 bear close resemblance to those in [8, Theorem 5], which characterizes s-structural controllability. When rephrased for observability by duality, [8, Theorem 5] coincides with Theorem 1 for the particular case where there is no unknown input in the system (1), and hence the input vertex set U in $\mathcal{H}$ and $\mathcal{H}_\times$ is empty.

Example 1 (continued). *Consider the bipartite graph $\mathcal{H}$ of Example 1, depicted in Fig. 3a. The following matching $\mathcal{M}$ is uniquely restricted and has size $5 = n + p$: $\mathcal{M} = \{(x_1, x_1'), (x_2, x_3'), (x_3, y_2), (u_1, y_1), (u_2, x_2')\}$.*

The bipartite graph $\mathcal{H}_\times$ is shown in Fig. 3b. In this example, $\mathcal{E}_{\text{loop}} = \{(x_1, x_1')\}$ and $\mathcal{E}_{\text{new}} = \{(x_2, x_2'), (x_3, x_3')\}$. The following matching $\mathcal{M}_\times$ is uniquely restricted, does not have any edge from $\mathcal{E}_{\text{loop}}$, and has size $5 = n + p$: $\mathcal{M}_\times = \{(x_1, y_2), (x_2, y_3), (x_3, x_3'), (u_1, y_1), (u_2, x_2')\}$.

We have shown that the structured system in Example 1 satisfies the two conditions i. and ii. of Thm. 1. Hence, this system is s-structurally unconstrained ISO.

3.3. Conditions for s-structural delay-1 left invertibility

In this section, we present some necessary and some sufficient conditions for s-structural delay-1 left-invertibility, and its characterization in the case with $D = 0$. The bipartite graphs involved in the statements are $\mathcal{N}$ and its subgraphs $\tilde{\mathcal{N}}$ and $\mathcal{N}_0$, defined as follows.

The bipartite graph $\mathcal{N}$ has left vertex set $U_0 \cup X \cup U_1$ and right vertex set $Y_0 \cup X' \cup Y_1$, where X and X' are the same as in the definition of $\mathcal{H}$, $U_0 = \{u_1^0, \ldots, u_p^0\}$ and $U_1 = \{u_1^1, \ldots, u_p^1\}$ are two copies of the input vertex set, and $Y_0 = \{y_1^0, \ldots, y_m^0\}$ and $Y_1 = \{y_1^1, \ldots, y_m^1\}$ are two copies of the output vertex set. The edge set is constructed as follows. For each index $h = 0, 1$, edges between U_h and Y_h are in one-to-one correspondence with non-zero entries of D: an entry $d_{ij} \neq 0$ corresponds to edges (u_j^0, y_i^0) and (u_j^1, y_i^1). Edges between U_0 and X' are in correspondence with non-zero entries of B, and edges between X and Y_1 are in correspondence with non-zero entries of C: an entry

$b_{ij} \neq 0$ corresponds to an edge (u_j^0, x_i') and an entry $c_{ij} \neq 0$ to an edge (x_j, y_i^1). Finally, all 'self-loops' (x_i, x_i') are added. Figure 2c shows a sketch of this construction, and Figure 3c illustrates $\mathcal{N}$ for Example 1.

We also define $\tilde{\mathcal{N}}$ as the subgraph of $\mathcal{N}$ obtained by removing the vertices in U_1 and all their incident edges, and $\mathcal{N}_0$ as the subgraph of $\tilde{\mathcal{N}}$ obtained by further removing the vertices in Y_0 and their incident edges.

We finally introduce the notation $\text{t-rk}(D)$ to denote the maximum rank that matrix D can attain, when the parameters in its non-zero entries can take arbitrary real values.

With the above-defined notation, we can study s-structural delay-1 left-invertibility. We start with some necessary conditions.

Theorem 2. *The following conditions are necessary for the structured system* (1) *to be s-structurally delay-1 left invertible:*

1. *The size of the maximum matching of $\mathcal{N}$ is $p + n + \text{t-rk}(D)$;*
2. *In $\tilde{\mathcal{N}}$ there exists a uniquely restricted matching of size $p + n$,*

where $p = |U_0|$ and $n = |X|$. ■

The following theorem gives a sufficient condition for s-structural delay-1 left invertibility.

Theorem 3. *The following condition is sufficient for the structured system* (1) *to be s-structurally delay-1 left invertible: in $\mathcal{N}$ there exists a uniquely restricted matching such that the set S of unmatched left vertices satisfies $S \subseteq U_1$ and there is no edge from S to any matched right vertex.* ■

Another sufficient condition for s-structural delay-1 left invertibility is the following, which was presented in [17] together with a simple direct proof of its sufficiency (see [17, Thm. 2]).

Proposition 1. *The following condition is sufficient for the structured system* (1) *to be s-structurally delay-1 left invertible: in $\mathcal{N}$ there exists a uniquely restricted matching of size $p + n + \text{t-rk}(D)$, where $p = |U_0|$ and $n = |X|$.* ■

At a first glance, none of the two conditions of Thm. 3 and Prop. 1 seems to imply the other. Indeed, both conditions are about the existence of a suitable uniquely restricted matching in $\mathcal{N}$, and they require different properties of such matching. Prop. 1 asks that such matching has size $p + n + \text{t-rk}(D)$, which is equivalent to asking that it covers all $U_0 \cup X$ and $\text{t-rk}(D)$ vertices of U_1. Theorem 3, instead, has the less stringent condition that the matching covers all $U_0 \cup X$, without any requirement on the number of matched vertices in U_1, but on the other hand it has the additional requirement that there is no edge from unmatched left vertices to matched right vertices.

In this paper we will show that the condition in Prop. 1 actually implies the one in Thm. 3, as stated below and proved in Sect. 8.2.

Proposition 2. *In the bipartite graph $\mathcal{N}$, if there exists a uniquely restricted matching of size $p + n + \text{t-rk}(D)$, where $p = |U_0|$ and $n = |X|$, then there exists a uniquely restricted matching such that the set S of unmatched left vertices satisfies $S \subseteq U_1$ and there is no edge from S to any matched right vertex.* ■

Clearly, thanks to Thm. 3, this provides an alternative proof of Prop. 1, different from the simpler one in [17]. The interest of this new proof is that it shows that there is no need to test both conditions separately: testing the condition in Theorem 3 is enough, since the class of systems that satisfy such condition includes all the systems that satisfy the condition in Prop. 1. We will show in Sect. 6 that Example 2 satisfies the condition in Theorem 3 and does not satisfy the condition in Prop. 1, thus showing that the latter is a more stringent condition than the former.

Recalling that ISO is equivalent to unconstrained ISO together with delay-1 left invertibility, each of the conditions in Theorem 1 and in Theorem 2 is necessary for s-structural ISO, and the the conditions in Theorem 1 and Theorem 3, together, are sufficient for s-structural ISO. Such conditions differ from the results in [15], where only the case $D = 0$ is considered, and further assumptions are made on matrices B and C. The case with general matrices is new for time-invariant systems. From the results in [16] about time-varying systems, we can only obtain a sufficient condition, not necessary ones. Indeed, asking the system to be ISO for all non-zero parameters, arbitrarily varying in time, is a more stringent request than only asking ISO to be satisfied for all constant non-zero parameters. The sufficient condition from [16, Thm. 2] involves a large graph, the so-called dynamic bipartite graph over a time interval of length n, whose number of vertices grows quadratically with n, while the results in Theorems 1, 2 and 3 involve smaller graphs, with order linear in n and comparable with the description of the LTI system. Moreover, the condition from [16, Thm. 2] turns out to imply the condition in Prop. 1, since the dynamic bipartite graph contains $\mathcal{N}$ as a subgraph, and the uniquely restricted matching mentioned in [16, Thm. 2] gives a uniquely restricted matching in $\mathcal{N}$ of size $p + n + \text{t-rk}(D)$ as required in Prop. 1. Hence, the sufficient condition from [16, Thm. 2] is actually more stringent than conditions in Thm. 1 and Thm. 3 together.

In general, there is a gap between the necessary conditions in Theorem 2 and the sufficient condition in Theorem 3, as we will show in Sect. 6. However, in the case where $D = 0$, i.e., there is no direct feedthrough of the input to the output, then there is no gap: the conditions in Theorem 2 are equivalent to the condition in Theorem 3, so that we have the following characterization of s-structural delay-1 invertibility.

Theorem 4. *If $D = 0$, then the structured system* (1) *is s-structurally delay-1 left invertible if and only if in the bi-*

partite graph $\mathcal{N}_0$ there exists a uniquely restricted matching of size $p+n$, where $p = |U_0|$ and $n = |X|$. ■

Proof: Since $D = 0$, the three bipartite graphs $\mathcal{N}$, $\tilde{\mathcal{N}}$, and $\mathcal{N}_0$ have the same edges and only differ by some stranded vertices, which are irrelevant for the existence of a uniquely restricted matching of some required size. Finally notice that $D = 0$ implies $\text{t-rk}(D) = 0$, so that the required size is $p + n$ both in the necessity and in the sufficiency part. □

Theorem 1 and Theorem 4, together, give the characterization of s-structural ISO for the systems without direct feedthrough of the input to the output ($D = 0$). This result encompasses a larger class of systems than the characterization of s-structural ISO in [15] (see Thm. 2 together with Remark 6), which required more stringent assumptions, not only $D = 0$, but also assumptions on B and C: each unknown input acts on a single state, no two inputs act on a same state, and dedicated sensors are available (i.e., each measurement concerns a single state).

Example 1 (continued). *Consider the bipartite graph $\mathcal{N}$ of Example 1, depicted in Fig. 3c. Consider the set of edges $\mathcal{L} = \{(u_1^0, y_1^0), (u_2^0, x_2'), (x_1, x_1'), (x_2, y_3^1), (x_3, y_2^1), (u_1^1, y_1^1)\}$. Notice that $\mathcal{L}$ is a uniquely restricted matching, and that the set of left vertices not covered by $\mathcal{L}$ is $S = \{u_2^1\} \subset U_1$. Hence, this system satisfies the sufficient condition for s-structural delay-1 left invertibility given in Thm. 3. Also notice that $\mathcal{L}$ is a maximum matching of $\mathcal{N}$. Hence, this system also satisfies the sufficient condition given in Prop. 1.*

We have obtained here that this structured system is s-structurally delay-1 left-invertible. Recalling that this system is also s-structurally unconstrained ISO (see Sect. 3.2), we obtain that it is s-structurally ISO.

4. Algorithmic versions of the graphical conditions

In this section we show that all graphical conditions in the main results from Sect. 3 can be tested in polynomial time. More precisely, most conditions can be tested with a complexity which is linear with respect to the total number of vertices and edges in the relevant bipartite graph; this corresponds to a linear complexity with respect to p, n, m (the sizes of input, state and output vector) and μ (the size of the parameter space, i.e., the total number of non-zero entries in matrices A, B, C, and D). We will further discuss how to rephrase the conditions involving color change rules, as in the literature on zero forcing sets.

4.1. Main algorithm and its properties

We present here the main algorithm to be used for testing the graphical conditions from Sect. 3. This algorithm takes as input a bipartite graph $\mathcal{B}$ (with left vertex set V, right vertex set W and edge set $\mathcal{E}$) and a subset of edges $\mathcal{F} \subseteq \mathcal{E}$. The output of the algorithm consists of a set of edges $\mathcal{M}$ and a set of left vertices S, whose properties are discussed in the remainder of this subsection. In Sect. 4.2 we will describe which input needs to be considered, so that the output of Algorithm 1 is informative about s-structural unconstrained ISO and delay-1 left invertibility.

Algorithm 1

Input: $\mathcal{B} = (V, W, \mathcal{E})$, $\mathcal{F}$.
Initialization: $\mathcal{M} = \emptyset$, $S = V$,
$T =$ the set of vertices in W having degree 1 and incident edge not in $\mathcal{F}$.
while $T \neq \emptyset$ **do**:
  Pick $w \in T$;
  $v =$ the unique neighbor of w;
  Add edge (v, w) to $\mathcal{M}$;
  Remove v from S;
  Remove v from $\mathcal{B}$ (i.e., remove v from V, and remove all edges incident to v from $\mathcal{E}$);
  $T =$ the set of vertices in W currently having degree 1 and incident edge not in $\mathcal{F}$.
end while
Return: S, $\mathcal{M}$.

In the remainder of this subsection, we will study the properties of the output of Algorithm 1, which make it relevant for testing s-structural (unconstrained) ISO.

Remark 1. *We consider the output $\mathcal{M}$, S of Algorithm 1. We denote by (v_1, w_1), ..., (v_s, w_s) the edges of $\mathcal{M}$. We can easily see that $\mathcal{M}$ is a matching, since vertices $v_1, \ldots, v_s$ and $w_1, \ldots, w_s$ are all distinct by construction. We also notice that $\mathcal{M}$ does not contain any edge from the set $\mathcal{F}$, i.e., $\mathcal{M} \cap \mathcal{F} = \emptyset$; for this reason, we will refer to $\mathcal{F}$ as the set of* forbidden *edges, since Algorithm 1 is forbidden to use such edges in the construction of the matching $\mathcal{M}$. Also, by construction, $S = V \setminus \{v_1, \ldots v_s\}$, i.e., S is the set of left vertices that are not covered by $\mathcal{M}$.*

In the case where there is more than one right vertex with degree 1 and incident edge not in $\mathcal{F}$, Algorithm 1 does not specify which of such vertices should be picked first; this choice can be made arbitrarily, as ensured by the following Proposition, whose proof is in Sect. 9.1.

Proposition 3. *Consider two runs of Algorithm 1, that might make different choices when picking a vertex $w \in T$, and let $\mathcal{M}_1$, S_1, and $\mathcal{M}_2$, S_2 be their outputs. Then $\mathcal{M}_1$ and $\mathcal{M}_2$ have the same cardinality, and $S_1 = S_2$.* ■

This property is important, because it shows that we can make statements about 'the output S of Algorithm 1', without the need to specify the rule used in Algorithm 1 for picking one among multiple right vertices with degree 1 and incident edge not in $\mathcal{F}$.

Proposition 4 below clarifies that Algorithm 1 finds a uniquely restricted matching, and more precisely the largest uniquely restricted matching with some given properties. See Sect. 9.1 for its proof.

Proposition 4. *The output $\mathcal{M}$ of Algorithm 1 is a matching satisfying the following three properties: 1) it is uniquely restricted, 2) $\mathcal{M} \cap \mathcal{F} = \emptyset$, and 3) there is no edge from unmatched left vertices to matched right vertices.*

S is the set of the left vertices not covered by $\mathcal{M}$.

If $\mathcal{M}'$ is another matching satisfying the three properties above, then $|\mathcal{M}'| \leq |\mathcal{M}|$ and the set S' of its unmatched left vertices satisfies $S \subseteq S'$.

In particular, $S = \emptyset$ if and only if there exists a uniquely restricted matching with no edge from $\mathcal{F}$ and covering all the left vertex set. ■

Notice that, in general, Algorithm 1 does not find the maximum uniquely restricted matching in $\mathcal{B}$ (even when setting $\mathcal{F} = \emptyset$), but rather finds the largest one that satisfies the above properties. Actually, the problem of finding the maximum uniquely restricted matching is NP-complete for bipartite graphs [19, Thm. 3.3]. Algorithm 1 (in a slightly different form discussed in Remark 4 in Sect. 5.2) has been used in [8, 9, 10] to test s-structural observability. Such test, rephrased for observability by duality, amounts to finding whether there is a uniquely restricted matching (without edges from a forbidden set $\mathcal{F}$) that covers all the left vertex set, in two suitable bipartite graphs. In this paper we have similar tests, and moreover we also need to consider a more general case, where not all left vertex set is covered.

4.2. Testing s-structural ISO using Algorithm 1

Thanks to Prop. 4, the conditions for s-structural ISO given in Sect. 3 can be tested using Algorithm 1, as we will present in details below. In particular, this means that they can be tested in polynomial time with respect to p, n, m (the sizes of input, state and output vector) and μ (the size of the parameter space, i.e., the total number of non-zero entries in matrices A, B, C, and D).

The characterization of s-structural unconstrained ISO given in Thm. 1 is equivalent to the following.

Theorem 5. *The structured system* (1) *is s-structurally unconstrained ISO if and only if the following conditions are satisfied:*

i. *Algorithm 1 with input $\mathcal{B} = \mathcal{H}$ and $\mathcal{F} = \emptyset$ returns $S = \emptyset$, and*
ii. *Algorithm 1 with input $\mathcal{B} = \mathcal{H}_\times$ and $\mathcal{F} = \mathcal{E}_{\text{loop}}$ returns $S = \emptyset$.* ■

Indeed, by Prop. 4 conditions i. and ii. in Thm. 1 are equivalent to the corresponding conditions in Thm. 5. As discussed in Appendix A, both conditions can be tested with complexity $O(p + n + m + \mu)$.

We then rephrase the necessary conditions for s-structural delay-1 left invertibility from Theorem 2.

Theorem 6. *The following conditions are necessary for the structured system* (1) *to be s-structurally delay-1 left invertible:*

1. *The size of the maximum matching of $\mathcal{N}$ is $p + n +$ t-rk(D), where $p = |U_0|$ and $n = |X|$, and*
2. *Algorithm 1 with input $\mathcal{B} = \tilde{\mathcal{N}}$ and $\mathcal{F} = \emptyset$ returns $S = \emptyset$.* ■

The first condition is the same as in Thm. 2, and the equivalence of the second condition with the correponding condition in Thm. 2 is obtained with Prop. 4.

The first condition in Thm. 6 can be checked with classical polynomial-time algorithms for finding the maximum matching, e.g., with Hopcroft-Karp algorithm [20], whose complexity is $O(\sqrt{p+n+m}\,(p+n+m+\mu))$. Testing the second condition has complexity $O(p + n + m + \mu)$, see Appendix A.

Finally, we turn our attention to the sufficient condition for s-structural delay-1 left invertibility given in Theorem 3, and to the characterization of s-structural delay-1 left invertibility for the case with $D = 0$ given in Theorem 4. Again, the reformulations given below are obtained using Prop. 4 and the given conditions can be tested in $O(p + n + m + \mu)$ as discussed in Appendix A.

Theorem 7. *The following condition is sufficient for the structured system* (1) *to be s-structurally delay-1 left invertible: Algorithm 1 with input $\mathcal{B} = \mathcal{N}$ and $\mathcal{F} = \emptyset$ returns $S \subseteq U_1$.* ■

Theorem 8. *If $D = 0$, then the structured system* (1) *is s-structurally delay-1 left invertible if and only if Algorithm 1 with input $\mathcal{B} = \mathcal{N}_0$ and $\mathcal{F} = \emptyset$ returns $S = \emptyset$.* ■

Concerning the sufficient condition given in Prop. 1 for s-structural delay-1 left invertibility, there is no need to test it: it is enough to test for the condition in Theorem 3. Indeed, Prop. 2 shows that the class of systems that satisfy the condition in Theorem 3 includes all the systems that satisfy the condition in Prop. 1. However, the interested reader can see in Sect. 8.1 that the same technique used to prove Prop. 2 also provides a technique for testing the condition in Prop. 1 in polynomial time, as described in Remark 5.

4.3. Color change rules on $\mathcal{G}$ and zero forcing sets

Two main approaches have been used in recent literature on s-structural controllability. On the one side, there are characterizations involving uniquely restricted matchings on a bipartite graph, together with algorithms similar to Algorithm 1, as discussed throughout this paper. On the other hand, literature on zero forcing sets presents characterizations involving the directed graph $\mathcal{G}$, with algorithms described as color change rules. At initialization, a subset $\mathcal{V}_{\text{black}} \subset V$ of vertices are colored black, and all other vertices are colored white; then the color change rule iteratively changes vertices from white to black under specified conditions, until no further color changes are possible. If the algorithm ends with all vertices being black, then $\mathcal{V}_{\text{black}}$ is called a *zero forcing set*. It has been shown that s-structural controllability is equivalent to the fact that the set of input vertices is a zero forcing set for suitable color change rules (see [12, Theorem 5.5] for a precise statement and its proof). The two approaches, with uniquely restricted matchings in bipartite graphs or with zero forcing sets in the directed graph, are equivalent, as first established in [12, Theorem 5.4].

In this section, we will show that most of our results can be easily rephrased with the use of suitable color change rules applied to the directed graph $\mathcal{G}$ or to a subgraph of $\mathcal{G}$.

Definition 5. *Given a directed graph, and given an initial coloring of its vertices with a subset $\mathcal{V}_{\text{black}}$ of vertices colored black and all other vertices colored white, we define the following two color change rules (the second rule also requires a given subset $\mathcal{V}_2$ of vertices, to which a further coloring rule applies).*

Color change rule n. 1: *If a vertex v is the unique white in-neighbor of a vertex w, then change v from white to black.*

Color change rule n. 2: *If a vertex v is the unique white in-neighbor of a black vertex w, or if $v \in \mathcal{V}_2$ is a white vertex with no white in-neighbor, then change v from white to black.*

The characterization of s-structural unconstrained ISO given in Thm. 1 (or its equivalent reformulation using Algorithm 1 given in Thm. 5) can be rephrased with the use of the two above-defined color change rules, as stated below and proved in Sect. 9.2.

Theorem 9. *The structured system* (1) *is s-structurally unconstrained ISO if and only if the following conditions are satisfied:*

i. *Color change rule n. 1 applied to the directed graph $\mathcal{G}$ with $\mathcal{V}_{\text{black}} = Y$ stops with all vertices being black, and*
ii. *Color change rule n. 2 applied to the directed graph $\mathcal{G}$ with $\mathcal{V}_{\text{black}} = Y$ and $\mathcal{V}_2 = X$ stops with all vertices being black.* ■

Expressing the conditions concerning delay-1 left invertibility with a color change rule on $\mathcal{G}$ is more involved: the bipartite graphs $\mathcal{N}$ and $\tilde{\mathcal{N}}$ used in Theorems 2 and 3 and in Prop. 1 have a double appearance of Y in the right vertex set, and this would require special care to distinguish their roles, defining a color change rule that separates the effect of edges from $\mathcal{E}_D$ and of edges from $\mathcal{E}_C$. Moreover, $\mathcal{N}$ also has a double appearance of U in the left vertex set, and this is even more difficult to be taken into account. One might devise a coloring associating two colors to each vertex of U, but this cumbersome rule will not be presented here. Another alternative is to use color change rule n. 1 on some directed graphs different from $\mathcal{G}$, such as the dynamic graph with vertices U_0, Y_0, X_1, U_1, Y_1 (see [21, Chapter 2] or [16] for a definition of the dynamic graph of a structured system) and a suitable subgraph of it.

In this paper, we will only consider the case $D = 0$, where the bipartite graph $\mathcal{N}_0$ involved in the characterization of delay-1 left invertibility in Theorem 4 has a single appearence of U and of Y and hence gives rise to a simple characterization involving a coloring rule on a subgraph of $\mathcal{G}$, as stated below and proved in Sect. 9.2.

Theorem 10. *Let $\mathcal{G}_0$ be the subgraph of $\mathcal{G}$ obtained by removing all edges between state vertices. In other words, $\mathcal{G}_0$ has vertex set $U \cup X \cup Y$ and edges corresponding to non-zero entries of B, C and D. If $D = 0$ (i.e., there are no edges from U to Y) then the structured system* (1) *is s-structurally delay-1 left invertible if and only if color change rule n. 2 applied to $\mathcal{G}_0$ with $\mathcal{V}_{\text{black}} = Y$ and $\mathcal{V}_2 = X$ stops with all vertices being black.* ■

5. From rank conditions to existence of suitable matchings

In this section we present the main tools that we will use in our proofs in Sections 7, 8 and 9, as well as in the study of examples in Sect. 6.

We start with known algebraic characterizations of (unconstrained) ISO that involve rank conditions, then we recall important relations between the rank of a matrix and suitable matchings in bipartite graphs, building upon the literature on structured systems [21], uniquely restricted matchings [18, 19] and s-structural controllability [7, 8].

5.1. Algebraic characterizations of (unconstrained) ISO

Both unconstrained ISO and delay-1 left invertibility have been studied in the systems theory literature, and there are well known algebraic characterizations of them, see e.g. [1, 2, 3]. Hereafter we will recall only the two characterizations that we will use in this paper, from [22, 23].

Lemma 1 ([22, Theorems 2.7 and 2.8]). *System* (1) *is unconstrained ISO if and only if the corresponding matrix pencil*

$$P(z) = \begin{bmatrix} A - zI & B \\ C & D \end{bmatrix}$$

has full column rank for all $z \in \mathbb{C}$. ■

Lemma 2 ([23, Prop. 2]). *System* (1) *is delay-1 left invertible if and only if $\text{rk}(Q) = p+n+\text{rk}(D)$, where p is the size of the input vector, n is the size of the state vector, and*

$$Q = \begin{bmatrix} D & 0 & 0 \\ B & -I & 0 \\ 0 & C & D \end{bmatrix}. \blacksquare$$

Remark 2. *When looking for a necessary condition for s-structural delay-1 left invertibility, we will also use the following immediate consequence of Lemma 2. We define $\tilde{Q}$ to be the submatrix of Q formed by its first $p+n$ columns, and we notice that $\text{rk}(\tilde{Q}) = p + n$ is a necessary condition for $\text{rk}(Q) = p + n + \text{rk}(D)$, and hence also for System* (1) *to be delay-1 left invertible.* ■

There is a fundamental connection between the above-defined matrices $P(z)$, Q and $\tilde{Q}$ and the bipartite graphs $\mathcal{H}$, $\mathcal{H}_\times$, $\mathcal{N}$, and $\tilde{\mathcal{N}}$ used in Sect. 3 to state our main results. We discuss the same in the following subsection.

5.2. Structured matrices and associated bipartite graphs

In this paper, we will use the following definitions of structured matrices and pattern matrices (note that a different vocabulary is used in some other papers). We say that a matrix M is a *structured matrix* if its entries are real polynomials in some variables, say $\lambda_1, \dots, \lambda_\mu$. We consider the variables as real-valued parameters. We say that a structured matrix M is a *pattern matrix* if all its non-zero entries are of the form λ_i, with all i's being distinct.

It is customary [21] to define the bipartite graph $\mathcal{B}(M)$ associated to the structured matrix M in the following manner: the left vertex set is the set of all columns of M, the right vertex set is the set of all rows of M, and there is an edge from column j to row i if and only if M_{ij} is not the all-zero polynomial. This associated bipartite graph plays a crucial role in various properties related to the rank of M.

The important connection between the matrices $P(z)$, Q and $\tilde{Q}$ from Sect. 5.1 and the bipartite graphs $\mathcal{H}$, $\mathcal{H}_\times$, $\mathcal{N}$, and $\tilde{\mathcal{N}}$ from Sect. 3 can now be highlighted.

Remark 3. *Matrices $P(z)$, Q and $\tilde{Q}$ are structured matrices. Their associated bipartite graphs are $\mathcal{B}(P(z)) = \mathcal{H}_\times$, $\mathcal{B}(Q) = \mathcal{N}$, and $\mathcal{B}(\tilde{Q}) = \tilde{\mathcal{N}}$. Moreover, when evaluated at $z = 0$, the matrix pencil $P(z)$ yields $P(0) = \left[\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right]$, which is a pattern matrix, whose associated bipartite graph is $\mathcal{B}(P(0)) = \mathcal{H}$.*

Another interesting remark is the following.

Remark 4. *When Algorithm 1 is applied to the bipartite graph $\mathcal{B}(M)$ associated with a structured matrix M, the algorithm can be equivalently rephrased by considering rows and columns of M instead of the corresponding right and left vertices of $\mathcal{B}(M)$. In this formulation, the algorithm takes as input a matrix M and a set of forbidden positions in the matrix. At each iteration, the algorithm looks for a row w with a unique non-zero entry, which moreover is not in a forbidden position; denoting by (w, v) the position of such unique non-zero entry, the algorithm then removes column v from the matrix. This rule for peeling off the columns of a matrix has been introduced in [8] as an algorithm for testing s-structural controllability, and used for the same purpose in subsequent works such as [9, 10].*

5.3. Rank of structured matrices

The rank of a structured matrix M clearly depends on the value of the parameters, but some relevant information about it can be obtained from the bipartite graph $\mathcal{B}(M)$.

A graphical notion that plays a relevant role is the one of term rank, see e.g. [21, Chapter 2]. The *term rank* of a structured matrix M, denoted $\text{t-rk}(M)$, is defined as the size of the maximum matching in $\mathcal{B}(M)$. For any choice of the parameters, $\text{rk}(M) \le \text{t-rk}(M)$. Moreover, if M is a pattern matrix, then $\text{rk}(M) = \text{t-rk}(M)$ generically, i.e., for all parameters except possibly a proper sub-variety of the parameter space. In particular, if M is a pattern matrix, then $\text{t-rk}(M)$ is the maximum rank that M can attain, when the parameters vary arbitrarily among real values. Notice that D is a pattern matrix, which justifies the notation $\text{t-rk}(D)$ introduced in Sect. 3.3.

Uniquely restricted matchings have been introduced in [18] to obtain the following result on the rank of a pattern matrix for all non-zero parameters.

Lemma 3 ([18, Thm. 3.9]). *If M is a pattern matrix, then the following are equivalent:*

- $\text{rk}(M) = r$ *for all non-zero parameters;*
- $\text{t-rk}(M) = r$ *and there exists a uniquely restricted matching of size r in $\mathcal{B}(M)$.* ■

A deeper understanding of the role of a uniquely restricted matching in the associated bipartite graph of a matrix will be given in the next subsection.

5.4. Uniquely restricted matchings and triangular submatrices

The following result gives a characterization of uniquely restricted matchings in bipartite graphs. This is the key tool that reveals the relation between uniquely restricted matchings and rank, and this is also the main tool to prove that the output $\mathcal{M}$ of Algorithm 1 is uniquely restricted.

Lemma 4 ([19, Thm. 3.1]). *Let $\mathcal{B}$ be a bipartite graph, with left vertex set V, right vertex set W, and edge set $\mathcal{E}$. A matching $\mathcal{M} \subseteq \mathcal{E}$ of size s is uniquely restricted if and only if there exists a reordering of vertices $V = \{v_1, \dots, v_{|V|}\}$ and $W = \{w_1, \dots, w_{|W|}\}$ such that $\mathcal{M} = \{(v_i, w_i)\}_{i=1}^s$ and moreover $(v_j, w_i) \notin \mathcal{E}$ for $1 \le i < j \le s$.* ■

When applying Lemma 4 to the bipartite graph $\mathcal{B}(M)$ associated with a structured matrix M, the relabeling of left vertices in the lemma corresponds to a permutation of columns of M and the relabeling of right vertices to a permutation of rows. More precisely, define two permutation matrices P_1 and P_2, where right multiplication of M by P_2 permutes the columns of M, moving the column corresponding to vertex v_j to position j, while left multiplication by P_1 permutes the rows, moving the row corresponding to vertex w_i to position i. With this notation, the square submatrix of $P_1 M P_2$ formed by its first s rows and columns has diagonal entries corresponding to the edges of the uniquely restricted matching $\mathcal{M} = \{(v_1, w_1), \dots, (v_s, w_s)\}$, i.e., $[P_1 M P_2]_{ii} = M_{w_i v_i} \neq 0$ for $i = 1, \dots, s$, and moreover it is lower triangular, since $(v_j, w_i) \notin \mathcal{E}$ for $1 \le i < j \le s$ means that $[P_1 M P_2]_{ij} = 0$ for $1 \le i < j \le s$.

Following the definition of Forms I, II and III in the literature (see e.g. [7]), we adopt the following definition from [15]: a matrix is in *Form IV* if it is equal to

$$\left[\begin{array}{ccccc} \times & & & & \\ & \times & & 0 & \\ & & \ddots & & \\ & * & & \times & \\ & & & & \times \\ \hline & & * & & \end{array}\right]$$

where $\times$ denote entries which are a non-zero polynomial, 0 denote entries equal to zero and $*$ denote unspecified entries, which might or might not be zero.

A uniquely restricted matching of $\mathcal{B}(M)$ covering all the left vertex set corresponds to P_1MP_2 being in Form IV, with the $\times$-terms corresponding to the edges in the uniquely restricted matching.

A uniquely restricted matching of $\mathcal{B}(M)$ of smaller size s corresponds to P_1MP_2 having two blocks of columns: a first block of s columns which is in Form IV, and then the remaining columns, whose entries are all unspecified ($*$), as follows

$$P_1MP_2 = \left[\begin{array}{ccccc|c} \times & & & & & \\ & \times & & 0 & & \\ & & \ddots & & & * \\ & * & & \times & & \\ & & & & \times & \\ \hline & & * & & & * \end{array}\right]$$

A uniquely restricted matching of $\mathcal{B}(M)$ such that there is no edge from unmatched left vertices to matched right vertices corresponds to P_1MP_2 as follows:

$$P_1MP_2 = \left[\begin{array}{ccccc|c} \times & & & & & \\ & \times & & 0 & & \\ & & \ddots & & & 0 \\ & * & & \times & & \\ & & & & \times & \\ \hline & & * & & & * \end{array}\right] \qquad (2)$$

where the zero block in upper-right position corresponds to the absence of edges from unmatched left vertices (columns $j > s$, corresponding to vertices in S) to matched right vertices (first s rows).

Clearly the above correspondences can be used in both the directions. Given a uniquely restricted matching of size s in $\mathcal{B}(M)$ (with no edge from unmatched left vertices to matched right vertices), after reordering it as in Lemma 4, we can use it to find permutation matrices P_1, P_2 such that the submatrix with the first s columns of P_1MP_2 is in Form IV (and the rest of the first s rows is zero). And, vice-versa, given permutation matrices P_1, P_2 such that the submatrix with the first s columns of P_1MP_2 is in Form IV (and the rest of the first s rows is zero) we can find a uniquely restricted matching of size s in $\mathcal{B}(M)$ (with no edge from unmatched left vertices to matched right vertices).

6. Conditions in Thm. 2, Thm. 3 and Prop. 1 are not equivalent.

In this section we present some examples that clarify the limitations of the sufficient and of the necessary conditions for s-strucutral delay-1 left-invertibility given in Sect. 3.3. Example 2 will highlight the gap between the more restrictive sufficient condition in Prop. 1 and the less restrictive one in Thm. 3, while Examples 3 and 4 will show the gap between the sufficient condition in Thm. 3 and the necessary condition in Thm. 2, by proving that the former is not necessary and the latter is not sufficient. Throughout this section, we will only define matrices B, C, and D for our examples. Indeed, matrix A is irrelevant for characterizing delay-1 left-invertibility, since it is not involved in the algebraic characterization in Lemma 2, nor in the construction of the bipartite graphs used in Thm. 2, Thm. 3 and Prop. 1.

We start by considering the two sufficient conditions for s-structural delay-1 left-invertibility given in Thm. 3 and Prop. 1. We have shown in Prop. 2 that the latter implies the former. We will now exhibit a structured system that satisfies the former but not the latter, thus showing that the sufficient condition in Thm. 3 encompasses a broader class of systems.

Example 2. *Consider the structured system* (1) *with any* 5×5 *matrix* A*, and matrices*

$$B = \begin{bmatrix} 0 & 0 & 0 \\ b_{21} & 0 & 0 \\ 0 & b_{32} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 0 & c_{13} & 0 & 0 \\ c_{21} & 0 & c_{23} & c_{24} & 0 \\ c_{31} & c_{32} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{44} & c_{45} \\ 0 & 0 & c_{53} & 0 & 0 \end{bmatrix},$$

$$D = \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & d_{32} & d_{33} \\ 0 & d_{42} & d_{43} \\ 0 & 0 & 0 \end{bmatrix}.$$

Its bipartite graph $\mathcal{N}$ *is shown in Fig. 4.*

Running Algorithm 1 with input $\mathcal{B} = \mathcal{N}$ *and* $\mathcal{F} = \emptyset$*, we obtain the following uniquely restricted matching of* $\mathcal{N}$*, covering all left vertices except* $S = \{u_2^1, u_3^1\} \subseteq U_1$*, and thus satisfying the sufficient condition for s-structural delay-1 left invertibility in Theorem 3:* $\mathcal{M}_1 = \{(u_1^0, y_1^0),$ $(x_1, x_1'),$ $(x_2, x_2'),$ $(x_4, x_4'),$ $(x_5, x_5'),$ $(x_3, y_2),$ $(u_2^0, x_3'),$ $(u_3^0, y_4^0),$ $(u_1^1, y_1^1)\}$ *(see edges highlighted in red in Figure 4).*

Now we can see that this system does not satisfy the more stringent sufficient condition for s-structural delay-1

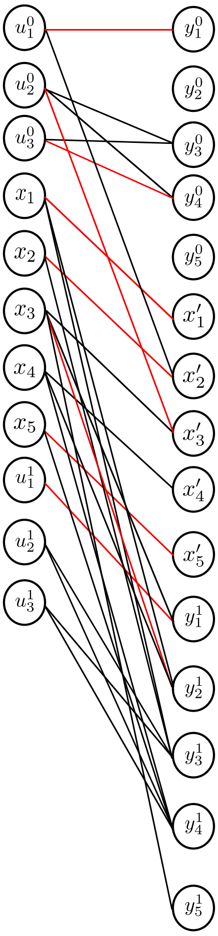
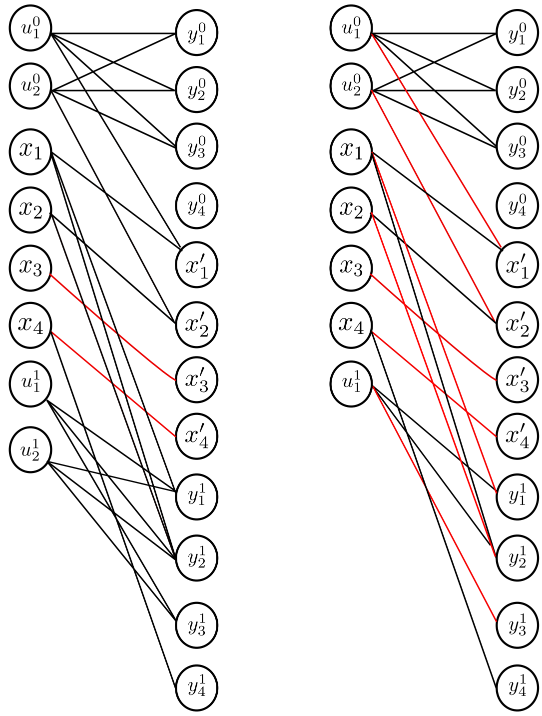


Figure 4: Bipartite graph $\mathcal{N}$ for Example 2. Red edges form a uniquely restricted matching $\mathcal{M}_1$.



(a) (b)

Figure 5: Bipartite graphs for Example 3: (a) $\mathcal{N}$, with in red the two edges of the uniquely restricted matching obtained with Algorithm 1; (b) $\mathcal{B}(\bar{Q})$, where $\bar{Q}$ is the submatrix of Q formed by the first seven columns; red edges form the uniquely restricted matching $\bar{\mathcal{M}}$.

left invertibility given in Prop. 1. Indeed, $\text{t-rk}(D) = 3$, *and hence* $p + n + \text{t-rk}(D) = 11$, *which equals the size of the left vertex set. This means that the condition in Prop. 1, applied to this system, requires the existence of a uniquely restricted matching covering all the left vertex set. By Prop. 4, since Algorithm 1 has returned* $S \neq \emptyset$, *such condition is not satisfied.*

The next two examples highlight the gap between the sufficient condition in Theorem 3 and the necessary conditions in Theorem 2. Example 3 below shows a system where the sufficient condition from Theorem 3 fails, but nevertheless the system is s-structurally delay-1 left invertible.

Example 3. *Consider the structured system* (1) *with any* 4×4 *matrix* A, *and matrices*

$$B = \begin{bmatrix} b_{11} & 0 \\ 0 & b_{22} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \; C = \begin{bmatrix} c_{11} & 0 & 0 & 0 \\ c_{21} & c_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{44} \end{bmatrix}, \; D = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \\ d_{31} & d_{32} \\ 0 & 0 \end{bmatrix}.$$

Its bipartite graph $\mathcal{N}$ *is shown in Fig. 5a.*

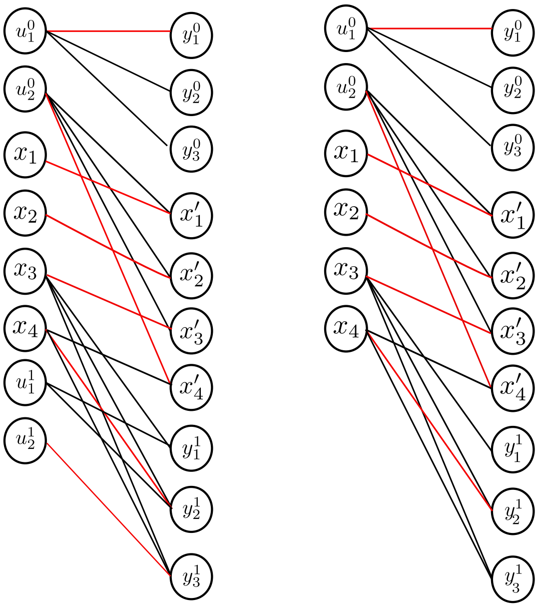
This system does not satisfy the sufficient condition for s-structural delay-1 left invertibility given in Thm. 3. Indeed, Algorithm 1, when running with input $\mathcal{B} = \mathcal{N}$ *and* $\mathcal{F} = \emptyset$, *can only remove left vertices* x_3 *and* x_4 *and then stops, with unmatched left vertices* $S = \{u_1^0, u_2^0, x_1, x_2, u_1^1, u_2^1\}$, *which is not a subset of* U_1 *and instead also includes elements of* U_0 *and of* X.

Now we will show that $\text{rk}(Q) = p + n + \text{rk}(D)$ *for all non-zero parameters, i.e., this system is s-structurally delay-1 left invertible. To do so, we will exploit the fact that* Q *is not a pattern matrix: some non-zero entries of* Q *cannot be chosen independently, since each of the parameters* $d_{11}, d_{12}, d_{21}, d_{22}, d_{31}, d_{32}$ *appears in two different positions.*

We will consider separately the two cases, where $\text{rk}(D) = 2$ *and* $\text{rk}(D) = 1$; *notice that* $\text{rk}(D)$ *cannot be zero with non-zero parameters.*

When $\text{rk}(D) = 2$, Q *is a block-lower-triangular matrix, and the blocks on the diagonal are* D, $-I$ *and* D, *each having full column rank. Hence,* $\text{rk}(Q) = 2 + 4 + 2 = p + n + \text{rk}(D)$.

When $\text{rk}(D) = 1$, *the second column of* D *is parallel to the first one; this also means that the eighth column of* Q *is parallel to the seventh one, and* $\text{rk}(Q) = \text{rk}(\bar{Q})$, *where* $\bar{Q}$ *is the submatrix of* Q *formed by the first seven columns only, without the eighth one. Considering the bi-*



(a) (b)

Figure 6: Bipartite graphs for Example 4: (a) $\mathcal{N}$; the edges in matching $\mathcal{M}_2$ are highlighted in red; (b) $\tilde{\mathcal{N}}$; the edges in matching $\mathcal{M}_3$ are highlighted in red.

partite graph $\mathcal{B}(\bar{Q})$ associated with $\bar{Q}$ (see Fig. 5b) and applying Algorithm 1 with input $\mathcal{B} = \mathcal{B}(\bar{Q})$ and $\mathcal{F} = \emptyset$, we can find the following uniquely restricted matching covering all the left vertex set of $\mathcal{B}(\bar{Q})$: $\bar{\mathcal{M}} = \{(x_3, x'_3), (x_4, x'_4), (u_1^1, y_3^1), (x_1, y_1^1), (x_2, y_2^1), (u_1^0, x'_1), (u_2^0, x'_2)\}$. As shown in Sect. 5.4, this is equivalent to the existence of permutation matrices P_1 and P_2 such that $P_1\bar{Q}P_2$ is in Form IV, and the $\times$-terms correspond to the edges of the matching $\bar{\mathcal{M}}$, so that they are equal to -1, -1, d_{31}, c_{11}, c_{22}, b_{11}, b_{22}. This shows that $\mathrm{rk}(\bar{Q}) = 7$ for all non-zero parameters, and hence also $\mathrm{rk}(Q) = 7 = p+n+\mathrm{rk}(D)$ for all non-zero parameters such that $\mathrm{rk}(D) = 1$.

The following example shows a system where the necessary conditions from Thm. 2 are satisfied, but this is not enough to ensure that the system is s-structurally delay-1 left invertible.

Example 4. *Consider the structured system* (1) *with any 4×4 matrix A and the following matrices B, C, and D:*

$$B = \begin{bmatrix} 0 & b_{12} \\ 0 & b_{22} \\ 0 & b_{32} \\ 0 & b_{42} \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 0 & c_{13} & 0 \\ 0 & 0 & c_{23} & c_{24} \\ 0 & 0 & c_{33} & c_{34} \end{bmatrix}, \quad D = \begin{bmatrix} d_{11} & 0 \\ d_{21} & 0 \\ d_{31} & 0 \end{bmatrix}.$$

The relevant bipartite graphs, $\mathcal{N}$ and $\tilde{\mathcal{N}}$, are depicted in Figures (6a) *and* (6b), *respectively.*

This example satisfies both necessary conditions for s-structural delay-1 left invertibility given in Theorem 2. For the first one, notice that $\text{t-rk}(D) = 1$ and the following matching in $\mathcal{N}$ has size $7 = p + n + \text{t-rk}(D)$: $\mathcal{M}_2 = \{(u_1^0, y_1^0), (u_2^0, x'_4), (x_1, x'_1), (x_2, x'_2), (x_3, x'_3), (x_4, y_2^1), (u_1^1, y_3^1)\}$. For the second one, the following matching is a uniquely restricted matching of $\tilde{\mathcal{N}}$ of size $6 = p+n$: $\mathcal{M}_3 = \{(u_1^0, y_1^0), (u_2^0, x'_4), (x_1, x'_1), (x_2, x'_2), (x_3, x'_3), (x_4, y_2^1)\}$.

This system is not s-structurally delay-1 left invertible. Here is an example of non-zero parameters such that $\mathrm{rk}(Q) = 6 < p + n + \text{t-rk}(D) = 7$:

$$B = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & -1 \end{bmatrix}, \quad D = \begin{bmatrix} -1 & 0 \\ -2 & 0 \\ 2 & 0 \end{bmatrix}.$$

7. Proofs of Theorems 1, 2, 3

These proofs are based on the algebraic characterizations from Lemmas 1 and 2. Since we are studying s-structural properties, the rank conditions are to be satisfied for all non-zero parameters. Such requirement will be turned into graphical conditions, using Lemma 3 (i.e., [18, Thm. 3.9]) and modified versions of its proof. The techniques to go beyond pattern matrices and study $P(z)$ are similar to the ones introduced in [7, 8] to characterize s-structural controllability. Delay-1 left invertibility requires particular care: differently from all results on s-structural controllability, Theorem 3 does not involve a uniquely restricted matching covering all the left vertex set. Its proof will use the discussion in Sect. 5.4, with a permuted matrix as in (2) instead of in Form IV as in previous results.

7.1. Proofs of necessity

In this subsection we prove the necessity part of Theorem 1 and Theorem 2.

For the necessity part of Theorem 1, we consider the algebraic characterization of unconstrained ISO from Lemma 1, which implies the following two necessary conditions for s-structural unconstrained ISO: $P(0)$ has full column rank for all non-zero parameters, and $P(1)$ has full column rank for all non-zero parameters. We will show that condition i. of Thm. 1 is necessary for the former, and condition ii. is necessary for the latter.

Recall that $P(0) = \left[\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right]$ is a pattern matrix, with associated bipartite graph $\mathcal{B}(P(0)) = \mathcal{H}$. Hence, by Lemma 3, it has full column rank $n+p$ if and only if there exists a uniquely restricted matching of size $n+p$ in $\mathcal{H}$.

For $P(1) = \left[\begin{smallmatrix} A-I & B \\ C & D \end{smallmatrix}\right]$, instead, we will use the Lemma 5 given below, which is a modified version of Lemma 3, inspired by [7, 8], and adapted to the presence of -1's in some entries of the matrix. In this lemma, we will consider the submatrices of $P(1)$ formed by selecting s columns of $P(1)$, and the associated bipartite graphs, which are the subgraphs of $\mathcal{H}_\times$ having left vertex set reduced to the s vertices corresponding to the selected columns. Lemma 5 gives the desired statement by setting $s = n + p$, but has a formulation with general s, more suitable for the proof by induction.

Lemma 5. *Let s be an integer, $1 \leq s \leq n+p$. For any matrix P_s formed with s columns of $P(1)$, if there does not exist a uniquely restricted matching $\mathcal{M}_\times^{(s)}$ of size s in $\mathcal{B}(P_s)$ that satisfies $\mathcal{M}_\times^{(s)} \cap \mathcal{E}_{\text{loop}} = \emptyset$, then either $P_s = 0$ or there exist non-zero parameters such that the corresponding numerical realization of P_s satisfies* $\text{rk}(P_s) < s$. ■

Proof: The proof is by induction on s. The base case is $s = 1$, where P_1 is a column of $P(1)$ (say the i-th column) and hence $\mathcal{B}(P_1)$ has only one vertex in its left vertex set. This implies that each edge is a matching of size 1, and is also a uniquely restricted matching. Hence, if there does not exist a uniquely restricted matching in $\mathcal{B}(P_1)$ with empty intersection with $\mathcal{E}_{\text{loop}}$, then all edges of $\mathcal{B}(P_1)$ are in $\mathcal{E}_{\text{loop}}$. This means that P_1 is either the all-zero vector, or a vector with a unique non-zero entry, which is equal to $a_{ii} - 1$ since the corresponding edge is in $\mathcal{E}_{\text{loop}}$. In the latter case, we can set $a_{ii} = 1$ and notice that this non-zero parameter corresponds to $P_1 = 0$, which implies $\text{rk}(P_1) = 0 < s = 1$.

Now we assume that the claim holds for $s-1$ (inductive assumption) and we prove that this implies that the claim holds for s. There are two cases, that require different proofs.

Case a): there exists a row of P_s having exactly one non-zero entry, and such that this non-zero entry is not of the form $a_{ii} - 1$, i.e., it is either a free parameter, or -1. Say that this is the k-th row, and its non-zero entry is in position (k, ℓ). Denote by P_{s-1} the submatrix of P_s obtained by removing the ℓ-th column. Notice that P_{s-1} is a matrix formed with $s-1$ columns of $P(1)$. If there is no uniquely restricted matching of size s in $\mathcal{B}(P_s)$ with empty intersection with $\mathcal{E}_{\text{loop}}$, then there is no uniquely restricted matching of size $s-1$ in $\mathcal{B}(P_{s-1})$ with empty intersection with $\mathcal{E}_{\text{loop}}$; indeed, if the latter existed, then one would obtain the former simply by adding the edge corresponding to the (k, ℓ)-th entry of P_s. Hence, by inductive assumption applied to P_{s-1}, either $P_{s-1} = 0$ or there exists some non-zero parameters such that the corresponding numerical realization of P_{s-1} has $\text{rk}(P_{s-1}) < s-1$. With the same parameters (if any), together with an arbitrary non-zero value for the (k, ℓ)-th entry in case it is a free parameter, we have a choice of non-zero parameters such that $\text{rk}(P_s) < s$.

Case b): the complement of case a). This means that all rows of P_s fall in the following categories: 1) all-zero row; 2) a row with $p \geq 1$ non-zero entries, one of which of the form $a_{ii} - 1$ for some i, and $p-1$ of which being free parameters; 3) a row with $p \geq 2$ non-zero entries, being either p free parameters, or one -1 and $p-1$ free parameters. If $P_s \neq 0$, we can find non-zero parameters such that the corresponding numerical evaluation P_s has all row-sums equal to zero. Rows in the first category already have zero sum. For rows in the second category, we can choose $a_{ii} = p$ and all other parameters (if any) equal to -1. For rows in the third category, we can choose one free parameter equal to $p-1$ and all other free parameters (if any) equal to -1. With this choice, the sum of the columns of P_s is the zero vector, which implies that $\text{rk}(P_s) < s$. □

Setting $s = n+p$ and recalling that $\mathcal{H}_\times = \mathcal{B}(P(1))$, Lemma 5 shows that condition ii. in Theorem 1 is necessary for s-structural unconstrained ISO.

Now consider Theorem 2. For the first part, we use the characterization of delay-1 left invertibility from Lemma 2: $Q = \left[\begin{smallmatrix} D & 0 & 0 \\ B & -I & 0 \\ 0 & C & D \end{smallmatrix}\right]$ has rank $p+n+\text{rk}(D)$. Recalling that $\text{rk}(Q) \leq \text{t-rk}(Q)$, a necessary condition for $\text{rk}(Q) = p+n+\text{rk}(D)$ is $\text{t-rk}(Q) \geq p+n+\text{rk}(D)$. Since D is a pattern matrix, $\text{rk}(D) = \text{t-rk}(D)$ generically, and hence in particular for some non-zero parameters; with these parameters for D, we have the following necessary condition for delay-1 left invertibility: $\text{t-rk}(Q) \geq p+n+\text{rk}(D) = p+n+\text{t-rk}(D)$. The proof is concluded by noting that $\text{t-rk}(Q)$ is the size of the maximum matching in $\mathcal{N} = \mathcal{B}(Q)$, and by noting that the inequality $\text{t-rk}(Q) \geq p+n+\text{t-rk}(D)$ can be re-written as an equality. Indeed, surely $\text{t-rk}(Q) \leq p+n+\text{t-rk}(D)$, because every matching of $\mathcal{N}$ can cover at most $\text{t-rk}(D)$ vertices of U_1, since its edges covering vertices of U_1 form a matching of $\mathcal{B}(D)$, whose size cannot exceed $\text{t-rk}(D)$.

For the second part of Theorem 2, recall the simple necessary condition for delay-1 left invertibility given in Remark 2: $\tilde{Q} = \left[\begin{smallmatrix} D & 0 \\ B & -I \\ 0 & C \end{smallmatrix}\right]$ must have full column rank $p+n$. Then, we can prove the following lemma, which gives the desired result when taking $s = p+n$.

Lemma 6. *Let s be an integer, $1 \leq s \leq p+n$. For any submatrix $\tilde{Q}_s$ formed with s columns of $\tilde{Q}$, if there does not exist a uniquely restricted matching of size s in $\mathcal{B}(\tilde{Q}_s)$, then either $\tilde{Q}_s = 0$ or there exist non-zero parameters such that the corresponding realization of $\tilde{Q}_s$ has* $\text{rk}(\tilde{Q}_s) < s$. ■

The proof by induction is the same as the proof of Lemma 5 in the simple case where $\mathcal{E}_{\text{loop}} = \emptyset$ and hence is omitted.

7.2. Proofs of sufficiency

In this subsection we prove the sufficiency part of Theorem 1 and Theorem 3.

The proof of the sufficiency part of Theorem 1 is inspired by the characterization of s-structural controllability [7, 8], and is based on the PBH-like characterization of unconstrained ISO in Lemma 1 and on the discussion in Sect. 5.4 about Form IV and uniquely restricted matchings.

We will show that condition i. in Theorem 1 ensures that $P(0)$ has full column rank $n+p$ for all non-zero parameters, and that condition ii. ensures that $P(z)$ has full column rank $n+p$ for all non-zero parameters and non-zero z, thus showing that the two together ensure s-structural left invertibility (by Lemma 1).

Condition i. is the existence in $\mathcal{H} = \mathcal{B}(P(0))$ of a uniquely restricted matching of size $n+p$ and hence covering all the left vertex set. As shown in Sect. 5.4 with the use of Lemma 4, this is equivalent to the existence of permutation matrices P_1 and P_2 such that $P_1P(0)P_2$ is in Form IV, with the $\times$-terms corresponding to the edges of the matching. Looking at the triangular square submatrix formed with the first $n+p$ rows of $P_1P(0)P_2$, it is clear that it has full rank $n+p$ for all non-zero parameters, since all diagonal entries (i.e., $\times$-entries) are parameters. Hence, also $P(0)$ has full column rank $n+p$ for all non-zero parameters.

Similarly, condition ii., i.e., the existence in $\mathcal{H}_\times = \mathcal{B}(P(z))$ of a uniquely restricted matching of size $n+p$ with no edge from $\mathcal{E}_{\text{loop}}$, is equivalent to the existence of permutation matrices P_1 and P_2 such that $P_1P(z)P_2$ is in Form IV, with the $\times$-terms corresponding to the edges of the matching, and hence being either free parameters or terms $-z$, but surely not zeros nor terms $a_{ii}-z$ since the matching does not use edges from $\mathcal{E}_{\text{loop}}$. This ensures that $P(z)$ has rank $n+p$ for all non-zero parameters and non-zero z.

The proof of Theorem 3 is also based on the discussion in Sect. 5.4, but involves a uniquely restricted matching covering only a suitable subset of the left vertex set.

Recall that $\mathcal{N} = \mathcal{B}(Q)$, and that by Lemma 2 delay-1 left invertibility is equivalent to $\text{rk}(Q) = p+n+\text{rk}(D)$.

Let $\mathcal{M} = \{(v_1, w_1), \dots, (v_{p+n+s}, w_{p+n+s})\}$ be a uniquely restricted matching as in the statement of Theorem 3, i.e., whose set of unmatched left vertices is $S \subseteq U_1$ and such that there is no edge from S to matched right vertices. We have used the notation s for the number of matched vertices in U_1; clearly $0 \le s \le r = \text{t-rk}(D)$. Assume $\mathcal{M}$ is already ordered as in Lemma 4. As discussed in Sect. 5.4, we can find permutation matrices P_1, P_2 such that P_1QP_2 has the first $p+n+s$ columns in Form IV, with the $\times$ entries equal to $Q_{w_1v_1}, \dots, Q_{w_{p+n+s}v_{p+n+s}}$ (in this order). The last $p-s$ columns of QP_2 are equal to the columns of Q corresponding to vertices in $S \subseteq U_1$; the last $p-s$ columns of P_1QP_2 are the same, except that some rows are permuted. The condition that there are no edges from S to matched right vertices ensures that the last $p-s$ columns of P_1QP_2 have the first $p+n+s$ rows equal to zero, i.e., we have

$$P_1QP_2 = \begin{bmatrix} Q_{11} & 0 \\ Q_{21} & Q_{22} \end{bmatrix},$$

with Q_{11} lower triangular and with diagonal entries equal to $Q_{w_1v_1}, \dots, Q_{w_{p+n+s}v_{p+n+s}}$; the last $p-s$ columns correspond to vertices $v \in S$. This implies that Q_{11} has full rank $p+n+s$ for all non-zero parameters, and hence $\text{rk}(Q) = p+n+s+\text{rk}(Q_{22})$ for all non-zero parameters.

Now consider the set $\tilde{\mathcal{M}}$ of edges in $\mathcal{M}$ that cover vertices of U_1 (in the same order with which they appear in $\mathcal{M}$). Notice that $\tilde{\mathcal{M}}$ is a uniquely restricted matching of $\mathcal{B}(D)$ of size s, with set of unmatched left vertices equal to S, and with no edge from S to unmatched right vertices. Hence, we have permutation matrices $\tilde{P}_1$, $\tilde{P}_2$ such that

$$\tilde{P}_1D\tilde{P}_2 = \begin{bmatrix} D_{11} & 0 \\ D_{21} & D_{22} \end{bmatrix},$$

where D_{11} is an $s \times s$ lower triangular matrix, with diagonal entries corresponding to the s entries $Q_{w_iv_i}$ with $v_i \in U_1$; the last $p-s$ columns correspond to vertices $v \in S$. This implies that D_{11} has full rank s for all non-zero parameters, and hence $\text{rk}(D) = s+\text{rk}(D_{22})$ for all non-zero parameters.

Now we will show that $\text{rk}(D_{22}) = \text{rk}(Q_{22})$. To see this, notice that the last $p-s$ columns of P_1QP_2 are obtained with row permutations from the columns of Q corresponding to vertices of U_1, i.e., from the columns of $\left[\begin{smallmatrix} 0 \\ 0 \\ D \end{smallmatrix}\right]$ corresponding to vertices of U_1. Similarly, the last $p-s$ columns of $\tilde{P}_1D\tilde{P}_2$ are obtained with row permutations from the columns of D corresponding to vertices of U_1. Hence, the last $p-s$ columns of P_1QP_2 and of $\tilde{P}_1D\tilde{P}_2$ have the same non-zero rows, and only differ in the number of zero rows and in their position. Moreover, Q_{22} contains all the non-zero rows of the last $p-s$ columns of P_1QP_2 and D_{22} contains all the non-zero rows of the last $p-s$ columns of $\tilde{P}_1D\tilde{P}_2$. This shows that Q_{22} and D_{22} have the same non-zero rows, and hence have the same rank.

Putting all these results together, for all non-zero parameters we have $\text{rk}(Q) = p+n+s+\text{rk}(Q_{22}) = p+n+s+\text{rk}(D_{22}) = p+n+\text{rk}(D)$, which concludes the proof of Theorem 3.

8. Maximum independent sets and proof of Prop. 2

8.1. Maximum independent sets and maximum matchings

A key tool that we will use to prove Prop. 2 is the well-known König theorem, that relates maximum independent sets and maximum matchings in bipartite graphs. We recall here this theorem and some consequences of it, which are useful both for the proof of Prop. 2 (see Sect. 8.2) and for testing in polynomial time whether a bipartite graph contains a uniquely restricted matching of the same size as its maximum matching (see Remark 5). The material in this subsection is mostly based on [18] and [24].

A maximum independent set I is a set of vertices such that there is no edge between any two of them, and with the largest size. For bipartite graphs, König theorem ensures that the complement K of a maximum independent set has the same size as the maximum matching. This also implies the following further properties.

Lemma 7. *Given a bipartite graph $\mathcal{B} = (V, W, \mathcal{E})$ denote by ρ the size of its maximum matching. Consider I a maximum independent set of $\mathcal{B}$, and define $K = (V \cup W) \setminus I$. Denote by $\mathcal{B}_{IK}$ the induced subgraph of $\mathcal{B}$ having left* *vertex set $V \cap I$ and right vertex set $W \cap K$ and by $\mathcal{E}_{IK}$ its edge set; similarly define $\mathcal{B}_{KK}$, $\mathcal{E}_{KK}$, $\mathcal{B}_{KI}$, and $\mathcal{E}_{KI}$. Then:*

(a) $|K| = \rho$;

(b) Every maximum matching of $\mathcal{B}$ is the disjoint union of a matching of $\mathcal{B}_{IK}$ of size $|V \cap K|$ and a matching of $\mathcal{B}_{KI}$ of size $|W \cap K|$;

(c) If $\mathcal{M}$ is a uniquely restricted matching of $\mathcal{B}$ of size ρ, then $\mathcal{M}$ is the disjoint union of a uniquely restricted matching of $\mathcal{B}_{IK}$ of size $|W \cap K|$ and a uniquely restricted matching of $\mathcal{B}_{KI}$ of size $|V \cap K|$;

(d) Given a uniquely restricted matching of $\mathcal{B}_{IK}$ of size $|W \cap K|$ and a uniquely restricted matching of $\mathcal{B}_{KI}$ of size $|V \cap K|$, their union is a uniquely restricted matching of $\mathcal{B}$ of size ρ. ■

Proof: Statement (a) is König theorem (see e.g. [25, Sect. 2.1]).

Statement (b) has been pointed out in [18, Theorem 3.1], and we recall here its short proof, adapted to our notation. Let $\mathcal{M}$ be a maximum matching. Since I is an independent set, $\mathcal{M}$ is the union of $\mathcal{M}_{KI}$, $\mathcal{M}_{KK}$, and $\mathcal{M}_{IK}$, which are matchings of $\mathcal{B}_{KI}$, $\mathcal{B}_{KK}$ and $\mathcal{B}_{IK}$, respectively. Notice that $|\mathcal{M}_{KK}| + |\mathcal{M}_{KI}| \leq |V \cap K|$, since both these matchings share the same left vertex set $V \cap K$, and similarly $|\mathcal{M}_{KK}| + |\mathcal{M}_{IK}| \leq |W \cap K|$, since both these matchings share the same right vertex set $W \cap K$. Hence, $2|\mathcal{M}_{KK}| + |\mathcal{M}_{KI}| + |\mathcal{M}_{IK}| \leq |K|$. On the other hand, $|\mathcal{M}_{KK}| + |\mathcal{M}_{KI}| + |\mathcal{M}_{IK}| = |\mathcal{M}| = |K|$. From these two expressions, we can conclude that $|\mathcal{M}_{KK}| = 0$ and $|\mathcal{M}_{KI}| + |\mathcal{M}_{IK}| = |K|$. Moreover, since $|\mathcal{M}_{KI}| \leq |V \cap K|$ and $|\mathcal{M}_{IK}| \leq |W \cap K|$, the latter equality also implies $|\mathcal{M}_{KI}| = |V \cap K|$ and $|\mathcal{M}_{IK}| = |W \cap K|$.

Finally, we obtain statements (c) and (d) as follows.

Given $\mathcal{M}$ a uniquely restricted matching of size ρ, define its decomposition $\mathcal{M} = \mathcal{M}_{KI} \cup \mathcal{M}_{IK}$ as above; this already shows that they have the desired size. Being submatchings of $\mathcal{M}$, they are uniquely restricted matchings of $\mathcal{B}$ and hence even the more so $\mathcal{M}_{KI}$ is uniquely restricted in $\mathcal{B}_{KI}$ and $\mathcal{M}_{IK}$ is uniquely restricted in $\mathcal{B}_{IK}$.

On the other hand, given $\mathcal{M}_{KI}$ a uniquely restricted matching of $\mathcal{B}_{KI}$ and $\mathcal{M}_{IK}$ a uniquely restricted matching of $\mathcal{B}_{IK}$, we can show with Lemma 4 that their union is a uniquely restricted matching of $\mathcal{B}$. Indeed, let $(v_1, w_1), \ldots, (v_h, w_h)$ be a reordering of $\mathcal{M}_{KI}$ such that $(w_j, v_i) \notin \mathcal{E}_{KI}$ (and hence also $(w_j, v_i) \notin \mathcal{E}$) for $1 \leq i < j \leq h$, and let $(v_{h+1}, w_{h+1}), \ldots, (v_{h+\ell}, w_{h+\ell})$ be a reordering of $\mathcal{M}_{IK}$ such that $(w_j, v_i) \notin \mathcal{E}_{IK}$ (and hence also $(w_j, v_i) \notin \mathcal{E}$) for $h+1 \leq i < j \leq h+\ell$. Then consider all edges $(v_1, w_1), \ldots, (v_{h+\ell}, w_{h+\ell})$. The absence of edges from $V \cap I$ to $W \cap I$ ensures that $(w_j, v_i) \notin \mathcal{E}$ for all $1 \leq i \leq h$ and $h+1 \leq j \leq h+\ell$, and hence we can conclude that this is a uniquely restricted matching for $\mathcal{B}$, since it has $(w_j, v_i) \notin \mathcal{E}$ for all $1 \leq i < j \leq h+\ell$. The size is $h + \ell = |V \cap K| + |W \cap K| = |K| = \rho$, which ends the proof. □

Remark 5. *Thanks to Lemma 7, the question whether a bipartite graph $\mathcal{B}$ has a uniquely restricted matching of the same size as its maximum matching can be answered as follows. First, find a maximum independent set I, and then test whether there exist:*

- *a uniquely restricted matching of $\mathcal{B}_{IK}$ covering all of its right vertices, and*
- *a uniquely restricted matching of $\mathcal{B}_{KI}$ covering all of its left vertices.*

Such test can be performed in polynomial time, as follows.

Thanks to Prop. 4, the existence of a uniquely restricted matching of $\mathcal{B}_{KI}$ covering all of its left vertices can be tested with Algorithm 1. Also the existence of a uniquely restricted matching of $\mathcal{B}_{IK}$ covering all of its right vertices can be tested with Algorithm 1, simply by exchanging the role of left and right vertices (in case $\mathcal{B}_{IK}$ is the bipartite graph associated with a matrix, this corresponds to looking at the transpose matrix). Algorithm 1 can be implemented with complexity $O(|V| + |W| + |\mathcal{E}|)$, as shown in Appendix A.

For the construction of a maximum independent set I in a bipartite graph, it is well-known that this can be done in polynomial time as follows. First, use Hopcroft-Karp algorithm [20] to construct a maximum matching $\mathcal{M}$; this has complexity $O(|\mathcal{E}|\sqrt{|V| + |W|})$. Then, use the following technique to construct a maximum independent set I associated with the maximum matching $\mathcal{M}$.

Let Z be the set that contains the unmatched left vertices, together with all vertices that can be reached by starting from an unmatched left vertex and following an alternating path. Then define $I = (V \cap Z) \cup (W \setminus Z)$ and define K as the complement of I. It is well known that I is a maximum independent set and K is a minimum vertex cover, see e.g. [25, Sect. 2.1], where this property is used to prove König Theorem. The construction of Z has complexity $O(|\mathcal{E}|)$ and then the construction of I and K has complexity $O(|V| + |W|)$. ■

Although presented differently, the test described in Remark 5 has been used in [18] to prove [18, Thm. 3.9] (presented here as Lemma 3), and in [24] to show that an answer can be found in polynomial time to the question whether a bipartite graph has a uniquely restricted matching of the same size as the maximum matching (see [24, Coroll. 4]).

8.2. Proof of Prop. 2

In order to prove Prop. 2, we turn our attention to the bipartite graph $\mathcal{N}$. We will use the short notation $r = \text{t-rk}(D)$ and V and W for the left and right vertex sets of $\mathcal{N}$, i.e., $V = U_0 \cup X \cup U_1$ and $W = Y_0 \cup X' \cup Y_1$.

We consider a uniquely restricted matching $\mathcal{M}$ of size $p+n+r$ in $\mathcal{N}$, which exists by assumption, and we define I to be the maximum independent set constructed from $\mathcal{M}$ as in Remark 5, i.e., Z is the set that contains the unmatched left vertices, together with all vertices that can be reached by starting from an unmatched left vertex and following an alternating path, and $I = (V \cap Z) \cup (W \setminus Z)$.

As a first part of the proof of Prop. 2, with the above-constructed I, we will now show that $I \cap V \subseteq U_1$.

Denote by $\mathcal{N}_1$ the induced subgraph of $\mathcal{N}$ having only left vertices in U_1 and right vertices in Y_1 (notice that $\mathcal{N}_1$ is isomorphic to $\mathcal{B}(D)$). Since $\mathcal{M}$ has size $p+n+r$, $\mathcal{M}$ covers all $U_0 \cup X$ and r vertices of U_1, and the r edges of $\mathcal{M}$ that cover vertices in U_1 form a maximum matching $\tilde{M}$ of $\mathcal{N}_1$. This implies $Z \cap V \subseteq U_1$. Indeed, assume by contradiction that there exists $v \in (U_0 \cup X) \cap Z$. This means that there exists an alternating path from some unmatched $u \in U_1$ to $v \in U_0 \cup X$; this path necessarily uses an odd number of edges from $\mathcal{N}_1$, and then uses an odd number of edges from $\mathcal{N} \setminus \mathcal{N}_1$. Consider the first portion of this path, formed with the edges from $\mathcal{N}_1$ only: its last vertex is a vertex $y \in Y_1$ covered in $\mathcal{M}$ with an edge (x, y) for some $x \in X$. This implies that y is not covered by $\tilde{M}$, and hence this first portion of path is an augmenting path for $\tilde{M}$ in $\mathcal{N}_1$, thus contradicting the fact that $\tilde{M}$ is a maximum matching of $\mathcal{N}_1$.

By definition of I, $Z \cap V \subseteq U_1$ means $I \cap V \subseteq U_1$.

We are now ready for the second and final part of the proof of Prop. 2: we construct a uniquely restricted matching in $\mathcal{N}$ whose unmatched left vertices belong to U_1, as follows. With I the above-constructed maximum independent set (such that $I \cap V \subseteq U_1$) and K its complement, we apply Lemma 7. We obtain that $\mathcal{M}$ is the disjoint union of two uniquely restricted matchings $\mathcal{M}_{IK}$ and $\mathcal{M}_{KI}$, where $\mathcal{M}_{KI}$ covers all vertices of $V \cap K = V \setminus I$ and hence the set of unmatched left vertices of $\mathcal{M}_{KI}$ is $V \cap I \subseteq U_1$. Hence, $\mathcal{M}_{KI}$ is the desired uniquely restricted matching which ends the proof of Prop. 2.

9. Proofs of results from Sect. 4

9.1. Proofs of properties of Algorithm 1

In this subsection we present the proofs of the properties of Algorithm 1 that were stated in Sect. 4.1.

In order to prove Prop. 3, we introduce a modified algorithm, which we will call Algorithm 2.

The only difference between Algorithms 1 and 2 is that one iteration of Algorithm 1 considers a single $w \in T$ (arbitrarily chosen), while one iteration of Algorithm 2 concerns all $w \in T$.

Proof of Prop. 3: To prove that any two runs of Algorithm 1 produce matchings of the same size and with the same set S of unmatched left vertices, we will show that any implementation of Algorithm 1 removes exactly the same left vertices that are removed by Algorithm 2. We will use the notation V_h for the set of left vertices removed at the hth iteration of Algorithm 2.

Algorithm 2

Input: $\mathcal{B} = (V, W, \mathcal{E})$, $\mathcal{F}$.
Initialization: $\mathcal{M} = \emptyset$, $S = V$.
T = the set of vertices in W having degree 1 and incident edge not in $\mathcal{F}$.
while $T \neq \emptyset$ **do**:
  for all $w \in T$ **do**
    v = the unique neighbor of w;
    Add edge (v, w) to $\mathcal{M}$;
    Remove v from S;
    Remove v from $\mathcal{B}$ (i.e., remove v from V, and remove all edges incident to v from $\mathcal{E}$);
  end for
  T = the set of vertices in W currently having degree 1 and incident edge not in $\mathcal{F}$.
end while
Return: S, $\mathcal{M}$.

Part 1: We show by induction that all left vertices removed by Algorithm 2 are also removed by Algorithm 1. The base case concerns V_1: we show that all vertices in V_1 will be removed also by Algorithm 1, sooner or later. Indeed, each $v \in V_1$ is the unique neighbor of some $w \in W$, with $(v, w) \notin \mathcal{F}$; this property is retained along iterations of Algorithm 1, and will remain true until v is removed from $\mathcal{B}$ because one of its neighbors (either w or another one) is picked by the algorithm; in either case, v must be removed before the while loop can terminate.

Then, we use the strong inductive assumption that all vertices from $V_1, \ldots, V_{h-1}$ are removed by Algorithm 1, and we show that then also vertices from V_h are removed. Consider Algorithm 1 at an iteration after all vertices from $V_1, \ldots, V_{h-1}$ are removed; now each $v \in V_h$ is either already removed, or it is the unique neighbor of some $w \in W$, with $(v, w) \notin \mathcal{F}$; again, this property is not destroyed along further iterations of Algorithm 1, and will remain true until v is removed, thus proving that v must be removed before the algorithm can terminate.

Part 2: We show the vice-versa: all vertices removed by Algorithm 1 are also removed by Algorithm 2. We show by induction that all vertices removed by Algorithm 1 belong to some V_h.

The first vertex removed by Algortihm 1 clearly belongs to V_1.

Then, when some vertex v_k is removed by Algorithm 1, by strong inductive assumption all vertices $v_1, \ldots, v_{k-1}$ removed by Algorithm 1 before v_k belong to $V_1 \cup \cdots \cup V_h$ for some h. This ensures that at iteration $h+1$ for Algorithm 2, since all vertices of $V_1 \cup \cdots \cup V_h$ have been removed, in particular $v_1, \ldots, v_{k-1}$ have been removed and hence either v_k has already been removed (i.e., belongs to $V_1 \cup \cdots \cup V_h$), or it now has degree 1 and unique incident edge not in $\mathcal{F}$, and hence belongs to V_{h+1}. □

Proof of Prop. 4: We consider the output $\mathcal{M}$, S of Algorithm 1. We denote by $(v_1, w_1), \ldots, (v_s, w_s)$ the edges of $\mathcal{M}$, in the order in which they are added to $\mathcal{M}$ by Algorithm 1 from first to last iteration.

As mentioned in Remark 1, $\mathcal{M}$ is a matching and S is the set of its unmatched left vertices. Also, $\mathcal{M}$ satisfies the property $\mathcal{M} \cap \mathcal{F} = \emptyset$.

Then, we notice that, for each $i = 1, \ldots, s$, we have $(v, w_i) \notin \mathcal{E}$ for all $v \notin \{v_1, \ldots, v_i\}$; indeed, at ith iteration of Algorithm 1, i.e., after $v_1, \ldots, v_{i-1}$ have been removed from $\mathcal{B}$, vertex w_i has a unique neighbor v_i. This remark implies two properties of the matching $\mathcal{M}$. First, by Lemma 4, $\mathcal{M}$ is uniquely restricted. Second, there is no edge from the set S of unmatched left vertices to the set $\{w_1, \ldots, w_s\}$ of matched right vertices. This ends the proof that $\mathcal{M}$ satisfies the three properties mentioned in Prop. 4.

Now consider $\mathcal{M}'$ any matching satisfying the same three properties, and let $(v'_1, w'_1), \ldots, (v'_h, w'_h)$ be its reordering as in Lemma 4. Notice that thanks to the properties of $\mathcal{M}'$ we can run Algorithm 1 by picking vertices $w'_1, \ldots, w'_h$, in this order. After these h iterations, Algorithm 1 has constructed a matching equal to $\mathcal{M}'$ and has a set $S' = V \setminus \{v'_1, \ldots, v'_h\}$ of unmatched left vertices. Then, either Algorithm 1 stops, and outputs $\mathcal{M}'$ and S', or continues for some more iterations, and returns $\mathcal{M}'' \supset \mathcal{M}'$ and $S'' \subset S'$.

We have shown in Prop. 3 that all runs of Algorithm 1 return matchings of the same size, and with the same set of unmatched vertices. This proves that $|\mathcal{M}| \geq |\mathcal{M}'|$ and $S \subseteq S'$, for any matching $\mathcal{M}'$ satisfying the three conditions.

Finally, notice that $S = \emptyset$ means that there is no unmatched left vertex, i.e., $\mathcal{M}$ covers all the left vertex set. In the absence of unmatched left vertices, the condition that there are no edges between unmatched left vertices and matched right vertices is trivially true. □

9.2. Proofs of Theorems 9 and 10

In this subsection, we prove the characterizations given in Sect. 4.3 using coloring rules, of s-structural unconstrained ISO and of delay-1 left-invertibility (the latter under the assumption $D = 0$).

Proof of Thm. 9: The key remark for this proof is that edges of $\mathcal{G}$ are in one-to-one correspondence with edges of $\mathcal{H}$. For vertices too there is a natural association between the vertex sets U and Y of $\mathcal{G}$ and the corresponding vertex sets in $\mathcal{H}$, while the vertex set X is doubly associated, both with the left vertex set X and with the right vertex set X', but the two roles can be easily distinguished by looking at outgoing and incoming edges, respectively. The out-neighbors of a vertex $v \in U \cup X$ in $\mathcal{G}$ are the same as the neighbors of the corresponding left vertex in $\mathcal{H}$, and similarly the in-neighbors of a vertex $w \in X \cup Y$ in $\mathcal{G}$ are the same as the neighbors of the corresponding right vertex in $\mathcal{H}$.

The first part of this proof is to show that the first condition in Thm. 9 is equivalent to the first condition in Thm. 5. To do so, we will show that an application of coloring rule n. 1 to $\mathcal{G}$ with $\mathcal{V}_{\text{black}} = Y$ corresponds to a run of Algorithm 1 with input $\mathcal{B} = \mathcal{H}$ and $\mathcal{F} = \emptyset$, in such a way that at each iteration the vertices in $\mathcal{G}$ that are white are in one-to-one correspondence with the left vertices of $\mathcal{H}$ that belong to S (i.e., that are currently present in $\mathcal{B}$, not having yet been removed).

This is clearly true at initialization: setting $\mathcal{V}_{\text{black}} = Y$, the set of white vertices of $\mathcal{G}$ is $U \cup X$. Then we construct a suitable run of Algorithm 1, in such a way that turning a vertex v from white to black in $\mathcal{G}$ corresponds to removing vertex v from S and from the left vertex set of $\mathcal{B}$. This is possible, because a vertex v is selected for a color change from white to black under the condition that it is the unique white in-neighbor of some vertex w, i.e., there exists some vertex $w \in X \cup Y$ such that w has exactly one white in-neighbor; looking at the corresponding left vertex v and right vertex w in $\mathcal{B}$, this means that v is the unique neighbor of w, and hence v can be removed in this iteration of Algorithm 1.

Since at each iteration the vertices of $\mathcal{G}$ that are white are in one-to-one correspondence with the left vertices of $\mathcal{B}$ that belong to S, the coloring rule terminates with no white vertices if and only if Algorithm 1 returns $S = \emptyset$.

The second part of this proof is to show that the second condition in Thm. 9 is equivalent to the second condition in Thm. 5, with a similar technique as in the first part of the proof. Here we consider an application of color change rule n. 2 to $\mathcal{G}$ with $\mathcal{V}_{\text{black}} = Y$ and $\mathcal{V}_2 = X$, and we construct a corresponding run of Algorithm 1 with input $\mathcal{B} = \mathcal{H}_\times$ and $\mathcal{F} = \mathcal{E}_{\text{loop}}$, in such a way that at each iteration the white vertices in $\mathcal{G}$ are in one-to-one correspondence with the remaining left vertices of $\mathcal{B}$; again, turning a vertex v from white to black in $\mathcal{G}$ corresponds to removing a vertex v from S and from $\mathcal{B}$. This is possible, since color change rule n. 2 is suitably tweaked so as to take care of the forbidden edges $\mathcal{F} = \mathcal{E}_{\text{loop}}$ and of the additional edge set $\mathcal{E}_{\text{new}}$ in $\mathcal{H}_\times$. Indeed, differently from rule n. 1, rule n. 2 does not allow w to be arbitrary: w must be black. First, notice that w being black ensures that w is different from v, and this, in turn, ensures that self-loops of $\mathcal{G}$ are avoided (equivalently: forbidden edges from $\mathcal{E}_{\text{loop}}$ are avoided). Second, notice that w being black is trivially true if $w \in Y$, and is a crucial assumption in case $w \in X$. Indeed, in case $w \in X$, and denoting by w and w' the left and right vertices of $\mathcal{B}$ corresponding to w, $v \neq w$ being the unique in-neighbor of w in $\mathcal{G}$ corresponds to two cases: when w is white, i.e., left vertex w has not yet been removed from $\mathcal{B}$ by Algorithm 1, w' has two neighbors v and w (the latter because of an edge from $\mathcal{E}_{\text{new}}$); when w is black, instead, w' has a unique neighbor v. Finally, the second part of color change rule n. 2 allows for white state vertices without any in-neighbor to be colored black. This corresponds to a pair of vertices x_i, x'_i in $\mathcal{B}$ that are neighbors with $(x_i, x_i') \in \mathcal{E}_{\text{new}}$ and such that x_i' has no other neighbor. □

Proof of Thm. 10: The proof is similar to the second part of the proof of Thm. 9. Notice that $\mathcal{G}_0$ has no self-loops (since it does not have any edges between state vertices), and hence $\mathcal{N}_0$ has a set of edges $\mathcal{K}_B \cup \mathcal{K}_C$ which is in one-to-one correspondence with the edge set of $\mathcal{G}_0$, plus a set of edges $\mathcal{K}_I$ containing all "self-loops" (x_i, x_i'). □

10. Conclusion

In the present paper, we studied s-structural (unconstrained) ISO for LTI network systems.

First, we provided a graphical characterization for s-structural unconstrained ISO. Then we gave some necessary and some sufficient conditions for s-structural ISO. We also compared our two sufficient conditions, and proved that one of the two implies the other, and the vice-versa is not true. Moreover, we showed with examples that in general there is a gap between the necessary and the sufficient conditions, while we proved that in the case without direct feedthrough from input to output there is no gap.

All our graphical conditions are in terms of existence of suitable uniquely restricted matchings in some bipartite graphs. We presented an algorithm to find uniquely restricted matchings with the required properties, and showed that its complexity is linear with respect to the size of the parameter space (total number of non-zero entries in the system matrices) plus the dimension of the input, state and output spaces. We also discussed the reformulation of our conditions as color change rules on the directed graph representing the network system, as in the literature on zero forcing sets.

One direction for future research is a deeper study of the case with direct feedthrough from input to output, aiming at closing the gap between necessary and sufficient conditions for s-structural ISO. Another future research line is the study of ISO with some known delay ℓ, for values of ℓ greater than 1; partial results in this direction have been obtained for structural ISO [26] and this problem is open for s-structural ISO.

Appendix A. Complexity of Algorithm 1

It is clear that Algorithm 1 has a complexity which is polynomial in $|V|+|W|+|\mathcal{E}|$. However, the actual complexity might vary depending on the implementation choices; in particular, the choice of how to represent the graph $\mathcal{B}$ affects the complexity of finding a vertex w with degree 1 and unique edge not in $\mathcal{F}$, and the complexity of removing its neighbor v from the graph. In [10], the authors consider the two specific applications of Algorithm 1 that are used to test s-structural controllability (that are somewhat similar to applying Algorithm 1 with $\mathcal{B} = \mathcal{H}$ and $\mathcal{F} = \emptyset$, and with $\mathcal{B} = \mathcal{H}_\times$ and $\mathcal{F} = \mathcal{E}_{\text{loop}}$), and they introduce a clever implementation whose complexity is linear in $|V| + |W| + |\mathcal{E}|$. The approach in [10] is based on two main ideas: using the compressed column storage (CSS) format for structured matrices, and storing in memory not only the main matrix but also its transpose and another structure relating the two.

We will now show that a complexity linear in $|V| + |W| + |\mathcal{E}|$ can be achieved for the general algorithm presented in this paper as Algorithm 1, and not only for its specific application used to test s-structural controllability. For simplicity of notation, we will only discuss the implementation details that are useful for achieving such complexity bound, while leaving to the interested reader the choice of the most efficient implementation of the data structures, for example with the CSS format proposed in [10]. We will present as Algorithm 3 an algorithm that is equivalent to Algorithm 1, and whose implementation choices give a complexity linear in $|V| + |W| + |\mathcal{E}|$.

Input representation:

The input of Algorithm 1 is: a bipartite graph $\mathcal{B} = (V, W, \mathcal{E})$ and a subset of edges $\mathcal{F} \subseteq \mathcal{E}$; edges in $\mathcal{F}$ can be interpreted as forbidden edges, since Algorithm 1 is forbidden to use edges from $\mathcal{F}$ in the construction of $\mathcal{M}$. We suppose that this input information is encoded in the following way:

- A structure `Edges` that contains, for each edge $e = (v, w) \in \mathcal{E}$: `leftvertex`$(e) = v$, `rightvertex`$(e) = w$, a flag `forbidden`$(e)$, which is `true` if $e \in \mathcal{F}$ and `false` if $e \notin \mathcal{F}$, and a flag `active`$(e) =$ `true`;
- A structure `Vertices` that contains, for each vertex $\nu \in V \cup W$, the list of all edges incident to ν.

This input representation is redundant: all the information describing $\mathcal{B} = (V, W, \mathcal{E})$ and $\mathcal{F}$ is contained in the structure `Edges` (except in the case where $\mathcal{B}$ has some stranded vertices, i.e., vertices without incident edges). However, the redundant structure `Vertices` is introduced, because it can be exploited to obtain a lower time complexity for the algorithm. Moreover, the flag `active` is trivially set to `true` for all edges, but it will be useful along iterations of the algorithm, to indicate when some edge is removed from the graph. Notice that, despite the redundancy, the memory requirement remains $O(|V| + |W| + |\mathcal{E}|)$ as in the most compact input description. Also notice that, given the structure `Edges`, a pre-processing step can construct the structure `Vertices` with $O(|V|+|W|+|\mathcal{E}|)$ operations.

Variables and their interpretation:

Algorithm 3 receives as input the structures `Edges` and `Vertices`. Along iterations, the flags `active` in the structure `Edges` are updated: changing `active`(e) from `true` to `false` corresponds to removing edge e from $\mathcal{B}$.

The other variables of Algorithm 3 are:

- `degree`, a vector of size $|W|$ with integer entries, that is initialized and updated in such a way that

$\texttt{degree}(w)$ equals the current degree of right vertex w, i.e., the number of its incident edges that are still active;

- $\mathcal{M}$ a list of edges, that is initialized as empty, and enriched with edges in such a way that at each iteration $\mathcal{M}$ is a matching with suitable properties (the same $\mathcal{M}$ as in Algorithm 1);
- S a vector of size $|V|$ with boolean entries, where $\texttt{S}(v) = \texttt{true}$ if v is an unmatched left vertex, i.e., a vertex $v \in V$ not covered by any edge from $\mathcal{M}$ (in the notation of Algorithm 1, $v \in S$);
- $\mathcal{T}$ a list of edges, that is initialized and updated in such a way that $\mathcal{T}$ is the list of all edges $e = (v, w)$ such that e is not forbidden and w currently has degree 1 (in the notation of Algorithm 1, such that $w \in T$).

Algorithm 3

```
Input: Edges, Vertices.
Initialization:
    M = ∅ ;
    for all v ∈ V, S(v) = true;   end for
    for all w ∈ W, degree(w) = number of edges incident in w;   end for
    T = the list of all edges e such that forbidden(e) = false and degree(rightvertex(e)) = 1.
while T ≠ ∅ do:
    Pick e ∈ T;
    Add e to M;
    v = leftvertex(e);
    S(v) = false;
    for all edge e' incident to v do
        active(e') = false;
        w' = rightvertex(e');
        degree(w') = degree(w') − 1;
        if degree(w') = 1 then
            for all edges ẽ incident to w' do
                if active(ẽ) = true and forbidden(ẽ) = false then add ẽ to T;
                end if
            end for
        end if
    end for
end while
Return: S, M.
```

Clearly, Algorithm 3 is equivalent to Algorithm 1. The action of removing a left vertex v in Algorithm 1 is obtained in Algorithm 3 by removing (or more precisely by flagging as not active) its incident edges; keeping the stranded vertex v with no incident edge, instead of removing it, has no impact on the rest of the algorithm. Similarly, picking $w \in T$ and letting v be its unique neighbor is obtained by picking $e = (v, w) \in \mathcal{T}$. The output of Algorithm 3 is the same as the one of Algorithm 1, other than the different way to represent the set S of unmatched left vertices (with a vector of flags, or with a list of vertices).

To study the complexity of Algorithm 3, we start by recalling that the extended input representation can be constructed in $O(|V| + |W| + |\mathcal{E}|)$. We can also show that the complexity of initialization is $O(|V| + |W| + |\mathcal{E}|)$. We notice that degree is initialized in $O(|W| + |\mathcal{E}|)$. Indeed, $\texttt{degree}(w)$ is obtained by counting the number of edges incident to w; this is done by looking at the list edges incident to w, from the structure Vertices, and requires $O(\deg w)$ for a vertex w with degree $\deg w \neq 0$ and $O(1)$ for a stranded vertex. Then recall that $\sum_{w \in W} \deg w = |\mathcal{E}|$. For the other initializations, it is clear that $\mathcal{M}$ is initialized in $O(1)$, S is initialized in $O(|V|)$, and $\mathcal{T}$ is initialized in $O(|\mathcal{E}|)$.

Then, we study iterations of Algorithm 3 (i.e., its 'while $\mathcal{T} \neq \emptyset$' cycle).

Recall that the output $\mathcal{M}$ is a matching of $\mathcal{B}$, and hence $|\mathcal{M}| \leq \min(|V|, |W|, |\mathcal{E}|)$. Notice that each edge that is added to $\mathcal{T}$ at some point (either at initialization, or at some iteration of the algorithm) is later removed from $\mathcal{T}$ and added to $\mathcal{M}$, and will never be added again to $\mathcal{T}$. This ensures that the action of adding an edge to $\mathcal{T}$ along iterations happens at most $|\mathcal{M}|$ times. Moreover, each iteration adds exactly one edge to $\mathcal{M}$, and hence the number of iterations is equal to $|\mathcal{M}|$.

Let us consider one iteration of the algorithm, where $e = (v, w)$ is removed from $\mathcal{T}$ and added to $\mathcal{M}$. We start by studying the complexity of such iteration, except for the part within the clause 'if $\texttt{degree(w')} = 1$'. We notice that each line requires $O(1)$ operations, thanks to the structures Edges and Vertices used to describe the graph $\mathcal{B}$. The cycle 'for all edge e' incident to v' involves $\deg v$ edges, where $\deg v$ denotes the degree of left vertex v, and the list of such edges can be read in $\texttt{Vertices}(v)$. Hence, the total cost of this iteration of the algorithm, except for the part in the 'if $\texttt{degree(w')} = 1$' clause, is $O(\deg v)$. We recall that each iteration corresponds to one edge of $\mathcal{M}$ and hence, denoting by $V_\mathcal{M}$ the set of left vertices covered by $\mathcal{M}$ (i.e., those that are flagged as false in the output S), we have that the cost of all iterations of the algorithm, except for their 'if $\texttt{degree(w')} = 1$' part, is $O(\sum_{v \in V_\mathcal{M}} \deg(v)) = O(\sum_{v \in V} \deg(v)) = O(|\mathcal{E}|)$.

Now we consider the 'if $\texttt{degree(w')} = 1$' part. If w' has degree 1, then we need to find the unique $\tilde{e}$ such that $\tilde{e}$ is incident to w' and is active, and in case it is not forbidden, we need to add $\tilde{e}$ to $\mathcal{T}$. To do so, we need to read the list of edges incident to w' and access the corresponding positions in the list of edges, to read the attributes active and forbidden; this costs $O(\deg w')$. For a given iteration of the algorithm, the 'if $\texttt{degree(w')} = 1$' clause might be true for no right vertex w', or for one, or for more. The key remark is that, along all iterations of the algorithm, this clause will be true only for some right vertices (those that are covered by the output matching $\mathcal{M}$

and that at initialization are not covered by an edge of $\mathcal{T}$), and only once for each of them. This means that the total cost, along all iterations, of the 'if $\texttt{degree(w}'\texttt{)} = 1$' parts is $O(\sum_{w' \in W} \deg w') = O(|\mathcal{E}|)$.

Putting together the above results, we have a total complexity $O(|\mathcal{E}|)$ for running all the iterations of the algorithm, and $O(|V| + |W| + |\mathcal{E}|)$ when including also the initialization and returning the output.

References

[1] H. L. Trentelman, A. A. Stoorvogel, M. L. J. Hautus, Control Theory for Linear Systems, Springer, 2001.

[2] N. Suda, E. Mutsuyoshi, Invariant zeros and input-output structure of linear, time-invariant systems, International Journal of Control 28 (4) (1978) 525–535.

[3] J. Massey, M. Sain, Inverses of linear sequential circuits, IEEE Trans. Computers C17 (4) (1968) 330–337.

[4] H. Mayeda, T. Yamada, Strong structural controllability, SIAM J. Control Optim. 17 (1) (1979) 123–138.

[5] W. Bachmann, Strenge strukturelle steuerbarkeit und beobachtbarkeit von mehrgrößensystemen/strong structural controllability and observability of multi-variable systems, at-Automatisierungstechnik 29 (1-12) (1981) 318–323.

[6] D. D. Olesky, M. Tsatsomeros, P. Van Den Driessche, Qualitative controllability and uncontrollability by a single entry, Linear Algebra and its Applications 187 (1993) 183–194.

[7] J. C. Jarczyk, F. Svaricek, B. Alt, Strong structural controllability of linear systems revisited, in: Proc. 50th IEEE Conf. Decision Control and European Control Conf. (CDC-ECC), Orlando, Florida, USA, 2011, pp. 1213–1218.

[8] A. Chapman, M. Mesbahi, On strong structural controllability of networked systems: a constrained matching approach, in: Proc. 2013 American Control Conf., Washington, DC, USA, 2013, pp. 6126–6131.

[9] G. Reißig, C. Hartung, F. Svaricek, Strong structural controllability and observability of linear time-varying systems, IEEE Trans. Autom. Control 59 (11) (2014) 3087–3092.

[10] A. Weber, G. Reißig, F. Svaricek, A linear time algorithm to verify strong structural controllability, in: Proc. 53rd IEEE Conf. Decision Control (CDC), Los Angeles, CA, USA, 2014, pp. 5574–5580.

[11] N. Monshizadeh, S. Zhang, M. K. Camlibel, Zero forcing sets and controllability of dynamical systems defined on graphs, IEEE Trans. Autom. Control 59 (9) (2014) 2562–2567.

[12] M. Trefois, J.-C. Delvenne, Zero forcing number, constrained matchings and strong structural controllability, Linear Algebra Appl. 484 (2015) 199–218.

[13] S. S. Mousavi, M. Haeri, M. Mesbahi, On the structural and strong structural controllability of undirected networks, IEEE Trans. Autom. Control 63 (7) (2018) 2234–2241.

[14] J. Jia, H. J. Van Waarde, H. L. Trentelman, M. K. Camlibel, A unifying framework for strong structural controllability, IEEE Trans. Autom. Control (2020) 1–1.

[15] S. Gracy, F. Garin, A. Y. Kibangou, Structural and strongly structural input and state observability of linear network systems, IEEE Trans. Control Netw. Syst. 5 (4) (2018) 2062–2072.

[16] S. Gracy, F. Garin, A. Y. Kibangou, Input and state observability of network systems with time-varying topology, IEEE Trans. Control Netw. Syst. 6 (2) (2019) 897–905.

[17] S. Gracy, F. Garin, A. Y. Kibangou, Strongly structural input and state observability for linear time invariant newtork systems, in: Proc. European Control Conference (ECC), Naples, Italy, 2019, pp. 2516–2521.

[18] D. Hershkowitz, H. Schneider, Ranks of zero patterns and sign patterns, Linear Multilinear Algebra 34 (1) (1993) 3–19.

[19] M. C. Golumbic, T. Hirst, M. Lewenstein, Uniquely restricted matchings, Algorithmica 31 (2) (2001) 139–154.

[20] J. E. Hopcroft, R. M. Karp, An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs, SIAM Journal on Computing 2 (4) (1973) 225–231. doi:10.1137/0202019.

[21] K. Murota, Matrices and matroids for systems analysis, Springer-Verlag, 2000.

[22] S. Sundaram, Fault-tolerant and secure control systems, University of Waterloo, Lecture Notes (2012).
URL `https://engineering.purdue.edu/~sundara2/misc/ft_control_lecture_notes.pdf`

[23] F. Garin, Structural delay-1 input-and-state observability, in: Proc. 56th IEEE Conf. Decision Control (CDC), Melbourne, Australia, 2017, pp. 2324–2329.

[24] L. D. Penso, D. Rautenbach, U. dos Santos Souza, Graphs in which some and every maximum matching is uniquely restricted, J. Graph Theory (2018) 1–9.

[25] R. Diestel, Graph Theory, 5th Edition, Vol. 173 of Graduate Texts in Mathematics, Springer-Verlag, Heidelberg, 2016.

[26] F. Garin, A. Y. Kibangou, Generic delay-l left-invertibility of structured systems with scalar unknown input, in: Proc. 58th IEEE Conf. Decision Control (CDC), Nice, France, 2019, pp. 3552–3556.