Connectivity Structure of Systems

Remco Bras
Bachelor Thesis
Department of Knowledge Engineering
Maastricht University
September 20, 2011

Abstract

In this paper, we consider to what degree the structure of a linear system is determined by the system’s input/output behavior. The structure of a linear system is a directed graph where the vertices represent the variables in the system and an edge \((x, y)\) exists if \(x\) directly influences \(y\). In a number of studies, researchers have attempted to identify such structures using input/output data. Thus, our main aim is to consider to what degree the results of such studies are valid. We begin by showing that in many cases, applying a linear transformation to a system will change the system’s graph. Furthermore, we show that even the graph’s components and their interactions are not determined by input/output behavior. From these results, we conclude that without further assumptions, very few aspects, if any, of a system’s structure are determined by its input/output relation. We consider a number of such assumptions. First, we show that for a number of parameterizations, we can characterize when two systems have the same structure. Second, in many applications, we can use domain knowledge to exclude certain interactions. In these cases, we can assume that a certain variable \(x\) does not influence another variable \(y\). We show that these assumptions cannot be sufficient to identify a system’s parameters using input/output data. We conclude that identifying a system’s structure from input/output data may not be possible given only assumptions of the form \(x\) does not influence \(y\).

1 Introduction

The aim of this paper is to consider to what degree the structure of a linear system can be determined using input/output data. By the structure of a linear system, we mean a graph in which the vertices represent the variables in the system and an edge from one variable to another exists if the second variable is directly influenced by the first one.

1.1 Motivation

This problem arose in a number of recent studies in which researchers have attempted to find the structure of a dynamical system using input/output data. For instance, researchers have attempted to find structure in the brain using fMRI data. In particular, Friston et al. propose the method of dynamic causal modeling. In this method, the activity of regions in the brain is modeled by a bilinear system. Each state variable \(z_i\) of this system represents the activity in a particular region of the brain. The change over time of this activity is given by

\[
\dot{z} = (A + \sum_{i=1}^{n_u} u_i B_i)z + Cu \tag{1}
\]

Here, \(u\) is the vector of inputs \(u_i\) to the model and \(n_u\) the number of such inputs. In dynamic causal modeling, these inputs represent experimental conditions. For some of these inputs, the corresponding column in the matrix \(C\) will be non-zero, implying that they affect state variables directly. For other inputs, the corresponding matrix \(B_i\) is non-zero, allowing these inputs to indirectly influence the state variables by changing how these variables affect each other.

In addition to the bilinear model describing brain activity, Friston et al. use what they call a “forward model” that describes how this activity is measured. This model depends on the particular measurement method used, such as fMRI or EEG. A forward model for fMRI measurements is given by Friston et al. [3].

To find structure in the brain, Friston et al. identify the parameters of both the bilinear model and the forward model. To do so, they assume the bilinear model has a particular form and the parameters that appear in this form and the forward model have certain given prior probability densities. Friston et al. then use the data to
find the posterior densities of the parameters. These densities can be used to make inferences about the parameters. For instance, by testing whether a particular entry of $A$ is non-zero, we can test if the data supports
the hypothesis that one variable influences another in the absence of input.

Goebel et al. [5] propose a different approach based on vector autoregressive (AR) models. In these models, a vector time-series $x_n$ is computed using its own past values, as shown in (2). Here, the integer $p$ is called the order of the vector AR model. The input $u_n$ is a stochastic white noise input with a given cross-covariance matrix. Unlike Friston et al., Goebel et al. do not use inputs based on experimental conditions.

$$x_n = -\sum_{i=1}^{p} A_i x_{n-i} + u_n \quad (2)$$

To quantify the influence of one variable on another, Goebel et al. use the concept of Granger causality. Goebel et al. distinguish two forms of this concept. The first of these is directed influence from one time series $x$ to a time series $y$. We say that $x$ causes $y$ if we can better predict $y_n$ using past values of both $x$ and $y$, that is, the set $S_{x,y} = \{ x_{n-1}, y_{n-1}, x_{n-2}, y_{n-2}, \cdots \}$, than using past values of $y$ alone, i.e. the set $S_y = \{ y_{n-1}, y_{n-2}, \cdots \}$. The second form of Granger causality is instantaneous causality between $x$ and $y$. This form of causality occurs if we can better predict $y$ from $S_{x,y} \cup \{ x_n \}$ than from $S_{x,y}$. Goebel et al. note that though the first form of causality is directed, the second is not.

To apply the concept of Granger causality to vector AR models, Goebel et al. create three such models. One model predicts $x$ in terms of its past values and another model performs a similar task for $y$. The third model uses past values of both $x$ and $y$ to predict both $x$ and $y$. Goebel et al. then use a number of measures based on the covariance matrices of the noise vectors that appear in these models to quantify the presence of Granger causality between $x$ and $y$.

The two papers we have considered so far focused on identifying structure in the brain from fMRI data. A similar method can also be applied in other fields. An example of this is the application described in Hollanders’ PhD thesis [6]. In this application, Hollanders considers time series of the expression levels of the genes of a unicellular fungus. Hollanders’ goal was to identify how these expression levels influence each other. To do so, Hollanders considers two model classes. One of these is the class of linear systems, the other a generalization of linear systems called piecewise linear systems. The state equation for one of Hollanders’ linear systems is given by (4). In this equation, $x(t)$ is the vector of gene expressions at time $t$, $u(t)$ is a vector of inputs and $\xi$ is a vector of Gaussian white noise. The matrices $A$ and $B$ are constant.

$$\dot{x}(t) = Ax(t) + Bu(t) + \xi_t \quad (3)$$

Hollanders’ second type of system, a piecewise linear system, is given by (4). Conceptually, this system consists of a set of $K$ linear subsystems as given by (4). At any given time $t$, the state is determined by subsystem $l(t)$. Hollanders assumes that the system switches between subsystems instantaneously. Therefore, $l(t)$ is constant in between two moments when the system switches from one subsystem to another.

$$\dot{x}(t) = A_{l(t)}x(t) + B_{l(t)}u(t) + \xi_t \quad (4)$$

To find the way in which gene expression levels influence each other, Hollanders uses a system identification approach to find a system of either the form (4) or one of the form (4) that fits the data. In order to find a unique solution, Hollanders’ approach uses a trade-off between minimizing an error criterion and criteria based on norms of the resulting system matrices.

A common feature of the papers we described above is that each paper proposes a method to find the structure of a dynamical system from input/output data. By the structure of a dynamical system, we mean a description of which variable directly influences another variable. This structure can be represented as a directed graph, in which each vertex represents a variable and the edge $(x, y)$ exists in the graph if the variable represented by $x$ directly influences the variable represented by $y$.

In the vector AR models of Goebel et al., we assume that the covariance matrix of the noise term is constant. This implies that the measures for Granger causality, which are based on these matrices, are constant for a given set of models. Therefore, the structure of a vector AR model as proposed by Goebel et al. will be constant.

For a linear system given by (4), one variable influences another if and only if a certain entry of either $A$ or $B$ is non-zero. Since the matrices $A$ and $B$ are constant, the structure of a linear system is also constant. This implies that the structure of each subsystem of a piecewise linear system is constant. Thus, a piecewise linear system must have one of a given set of structures at any moment. This structure changes only when the system switches from one subsystem to another.

In bilinear systems such as those used by Friston et al., the structure of the system is determined by the input signal. Therefore, the structure can change from moment to moment, dictated by changes in input. The way in which the inputs dictate the structure is, however, constant.

(v. September 20, 2011, p.2)
From the above discussion, we see that for a given system, either the structure is constant from moment to moment or a description of the possible structures exists. For the identification procedures of Goebel et al., Friston et al. and Hollander to be practically useful, the possible structures a system can have should be uniquely determined by input/output data. As mentioned, the goal of this paper is to describe to what degree this is the case for a linear system.

1.2 Related work
Before we consider how we will proceed and state our main conclusions, we note that related work has been done in the area of linear structured systems. Essentially, a linear structured system as used by Dion et al. and other researchers is identical to what we call the structure of a given linear system. The question considered by researchers in this area is what information about systems with a particular structure can be derived from the structure itself. As Dion et al. indicate, a number of properties will either not hold at all for systems with a given structure or hold for almost all systems with this structure. A property that holds for most systems with a structure is said to be generic for this system. Dion et al. give graph-theoretical criteria for genericity of a number of properties, such as controllability, observability and the solvability of a number of other control problems. Though this work is useful, it does not answer our basic question, namely how a given input/output relation determines the structure of systems realizing this relation.

1.3 Contribution
In this paper, we consider to what degree the structure of a linear system can be determined using the input/output data it generates. First, we consider how linear transformations affect the structure of a linear system. Here, we see that for many linear systems, there exists a linear transformation that changes the structure of the system. Unless the linear transformation is part of a relatively small set, we have no guarantee that it does not change the structure of the system.

In light of this result, we consider whether a weaker graph-theoretical relation than isomorphism could allow us to find some aspect of the structure that is conserved by linear transformations. For this purpose, we consider a number of variants of graph homomorphism. Unfortunately, each variant is either not an equivalence relation or leads to strange results where systems that should have different structures have the same structure.

Given that linear transformations that result in isomorphic graphs appear to be rare and variants of homomorphism lead to undesirable results, we apply graph isomorphism to condensed graphs. The condensed graph of a system is obtained from the original graph of the system by replacing each strong component by a single vertex. An edge exists from one component to another if a vertex in the first component had an edge to a vertex in the second component in the original graph. Unfortunately, the same problem we saw with graph isomorphism applied directly to a system’s graph also occurs with condensed graphs. That is, there exist linear transformations that result in systems with different condensed graphs.

The results discussed above indicate that if we do not make any assumptions about the system, very few, if any, aspects of the system’s structure are determined by input/output data. Thus, it seems that we need to make assumptions about a system in order to determine its structure using this data.

The first kind of assumption we consider is that the system is a minimal SISO system in some canonical form. In a number of special cases, these assumptions allow us to characterize the existence of isomorphisms between the graphs of two systems. In other words, we can characterize when two systems have the same structure.

The second kind of assumption we consider is that some variables do not influence certain other variables. That is, we assume that some edges do not occur in the graph of our system. This implies that the graph of our system must be a subgraph of a given graph. Using this graph, a number of properties of systems satisfying our assumptions can be derived, as described by Dion et al. Our first result concerning these properties is that unless there is no edge from a state variable to an output in the given graph, we cannot uniquely identify our system from input/output data. This implies that for a realistic system, where the state variables do influence the output, we cannot find the system parameters using input/output data. In our second result, we give a graph-theoretical characterization of graphs such that almost all systems satisfying the assumptions given by this graph are minimal. These conditions are also necessary conditions for any given system to be minimal.

1.4 Application of the results
Since the results described above apply only to linear systems, they are not directly applicable to the models considered by Friston et al. Goebel et al. and Hollander. However, the linear systems we consider are strongly related to each of the model classes used by these researchers. For instance, if we set each matrix $B_i$ to zero in (1), we find the state equation of a linear system. Thus, we may identify the resulting system with a linear system where the output of the system equals its state. Due to the assumption that the system’s output equals its state, we cannot directly apply a linear transformation to such a system. If we instead allow the output to be any linear transformation of the state,
we find a subset of the bilinear systems corresponding to the linear systems such that the input does not directly influence the output. Many of our results apply to this set of linear systems. Since this set of systems is a proper subset of the set of all bilinear systems, we conjecture that these results may apply, possibly in a modified form, to the class of all bilinear systems.

A similar relation exists between the linear systems we consider and the piecewise linear systems used by Hollander. Indeed, if we ignore the noise term in (4) and allow the output of the system to be a linear combination of its state, we find the same set of linear systems we considered above. In this case, we can generalize to piecewise linear systems by allowing the system to have an arbitrary number of subsystems. Since the systems with 1 subsystem are a subset of this more general class, we conjecture that many of our results may also hold for this more general case.

The relation between the vector AR models considered by Goebel et al. and our linear systems is more complex. Though we can represent each vector AR model of the form (2) by a linear system driven by a stochastic input, our notion of this system’s graph structure may differ from the Granger causality criteria used by Goebel et al. In some cases, these structures can give equivalent results. For instance, suppose we have two time series $x_i \in \mathbb{R}^m$ and $y_i \in \mathbb{R}^n$. Suppose the best vector AR model of $x_i$ and $y_i$ of the form (2) has block-diagonal matrices $A_i$, where $A_i = \begin{bmatrix} A_{i,x} & 0 \\ 0 & A_{i,y} \end{bmatrix}$, $A_{i,x} \in \mathbb{R}^{m \times m}$ and $A_{i,y} \in \mathbb{R}^{n \times n}$, and the covariance matrix of the noise $u_n$ is also block-diagonal, i.e. this matrix has the form $\begin{bmatrix} \Sigma_x & 0 \\ 0 & \Sigma_y \end{bmatrix}$, where $\Sigma_x \in \mathbb{R}^{m \times m}$ and $\Sigma_y \in \mathbb{R}^{n \times n}$. In this case, the graph of this system consists of two disjoint subgraphs, one for the vector AR model of $x_i$ and one for the model of $y_i$. Since the joint model of $x_i$ and $y_i$ predicts each of these time series using only its own past values, we also find that there is no Granger causality between $x_i$ and $y_i$. Thus, we see that the graph structure and Granger causality criteria can give equivalent results. It is unclear at this point whether and, if so, how, this holds in the general case.

The remainder of this paper is structured as follows. In Section 2, we recall the definition of a linear system, its associated graph and other relevant concepts. We then use these concepts in Section 3 to state the results we have discussed above. These results are proved in Section 4. In the final section, we briefly review our main results and their implications.

2 Systems, graph structure and equivalent structures

In this section, we recall the definition of a discrete-time LTI system and its associated graph structure. Given an initial state $x_0$ and an input sequence $u_k$, the evolution over time of the state $x_k$ and output $y_k$ of the system is given by (4) and (5). In these equations, the matrices $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, $C \in \mathbb{R}^{n_y \times n_x}$ and $D \in \mathbb{R}^{n_y \times n_u}$ are constants. Since these constants uniquely determine the discrete-time LTI system, we define such a system to be a 4-tuple of the matrices $(A, B, C, D)$, as in Definition 1 below. Similar definitions can be found in many textbooks on linear systems, such as those by Vaccaro [3] and Kailath [4].

$$x_{k+1} = Ax_k + Bu_k$$
$$y_k = Cx_k + Du_k$$

Definition 1. An LTI state-space system with $n_u$ inputs, $n_y$ outputs and $n_x$ state variables is a 4-tuple of matrices $(A, B, C, D)$, where $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, $C \in \mathbb{R}^{n_y \times n_x}$, $D \in \mathbb{R}^{n_y \times n_u}$.

We define the structure of a state-space system as a directed graph whose vertices are the inputs, outputs and state variables of the system. An edge from one variable to another variable exists in this graph if $y_i$ is directly influenced by $x_i$. We formalize this concept below. We begin by defining the graph of a matrix and then use this concept to define the graph of a system. Before we state these definitions, we briefly recall the concept of a directed graph as defined in many textbooks on graph theory, such as the book by Chartrand et al. [1, Ch. 7].

Definition 2. A (directed) graph $G$ is a 2-tuple $(V,E)$, where $V$ is a finite set of objects called vertices and $E \subset V \times V$ is a set of 2-tuples of vertices called edges.

Definition 3. Let $M \in \mathbb{R}^{n \times n}$. Then, the associated graph $G$ of the matrix $M$ is given by $G = (V,E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ and $E = \{(v_i, v_j) | M_{ji} \neq 0\}$.

Definition 4. The associated graph of the system $S = (A, B, C, D)$ with $n_u$ inputs, $n_x$ state variables and $n_y$ outputs is defined as the graph $G(S) = (V,E) = (\{v_1, v_2, \ldots, v_n\}, E)$ of the matrix $M_S = \begin{bmatrix} A & B \\ 0 & C \end{bmatrix}$, where $n = n_x + n_u + n_y$ and $M_S \in \mathbb{R}^{n \times n}$. Additionally, we will use the following notation. The vertices $v_1$ through $v_{n_x}$ are called the state variable vertices of $G(S)$. We will denote these vertices by $v_i(G(S)) = v_1$. The vertices $v_{n_x+1}$ through $v_{n_x+n_u}$ are called the input vertices of $G(S)$, denoted by $u_i(G(S)) = v_{n_x+i}$. Finally, the remaining vertices are called the output vertices $y_i(G(S))$, given by $y_i(G(S)) = v_{n_x+n_u+i}$.
Remark 1. Unless otherwise noted, \( G(S) \) denotes the graph of a linear system as defined in Definition \([4]\). When the graph we are referring to is clear from the context, we will abbreviate \( x_i(G(S)) \) to \( x_i \). We will similarly abbreviate \( y_i(G(S)) \) to \( y_i \) and \( u_i(G(S)) \) to \( u_i \).

A useful corollary of Definition \([4]\) is that only certain edges occur in the graph \( G(S) \). Furthermore, these edges occur if and only if the corresponding entries in the system matrices of \( S \) are non-zero. We state this formally below.

**Corollary 1.** Let \( S = (A, B, C, D) \) be a linear system with \( n_u \) inputs, \( n_x \) state variables and \( n_y \) outputs. Then, in the graph \( G(S) = (V, E) \) of \( S \), each edge \((v_1, v_2)\) is of one of the forms listed below.

1. \((x_i, x_j)\), where \( 1 \leq i, j \leq n_x \)
2. \((u_i, x_j)\), where \( 1 \leq i \leq n_u \wedge 1 \leq j \leq n_x \)
3. \((x_i, y_j)\), where \( 1 \leq i \leq n_x \wedge 1 \leq j \leq n_y \)
4. \((u_i, y_j)\), where \( 1 \leq i \leq n_u \wedge 1 \leq j \leq n_y \)

In addition, the following conditions hold.

1. \((x_i, x_j) \in E\) if and only if \( A_{ij} \neq 0 \)
2. \((u_i, x_j) \in E\) if and only if \( B_{ij} \neq 0 \)
3. \((x_i, y_j) \in E\) if and only if \( C_{ij} \neq 0 \)
4. \((u_i, y_j) \in E\) if and only if \( D_{ij} \neq 0 \)

In this paper, we will consider a number of functions from the vertex set of one graph to the vertex set of another. We will restrict these functions to only map vertices onto vertices "of the same kind". That is, inputs must be matched onto inputs, outputs onto outputs and so on. Formally, we say that the function must be type-restricted, as defined in Definition \([6]\) below.

**Definition 5.** Let \( S_1 \) be a linear system with \( n_{x,1} \) state variables, \( n_{u,1} \) inputs and \( n_{y,1} \) outputs and \( S_2 \) a linear system with \( n_{x,2} \) state variables, \( n_{u,2} \) inputs and \( n_{y,2} \) outputs. Then, the vertices \( v_1 \in V(G(S_1)) \) and \( v_2 \in V(G(S_2)) \) are of the same type if one of the following conditions holds for some \( i \) and \( j \):

1. \( 1 \leq i \leq n_{u,1} \wedge 1 \leq j \leq n_{u,2} \wedge u_i(G(S_1)) = v_1 \wedge u_j(G(S_2)) = v_2 \)
2. \( 1 \leq i \leq n_{x,1} \wedge 1 \leq j \leq n_{x,2} \wedge x_i(G(S_1)) = v_1 \wedge x_j(G(S_2)) = v_2 \)
3. \( 1 \leq i \leq n_{y,1} \wedge 1 \leq j \leq n_{y,2} \wedge y_i(G(S_1)) = v_1 \wedge y_j(G(S_2)) = v_2 \)

**Definition 6.** Let \( S_1 \) and \( S_2 \) be linear systems and let \( G(S_i) = (V_i, E_i) \) for \( i = 1, 2 \). A type-restricted function \( \phi : V_1 \to V_2 \) is a type-restricted isomorphism if:

1. \( \phi \) is a bijection, that is, it is both injective and surjective.
2. For any vertices \( u, v \in V_1 \), \((u, v) \in E_1\) if and only if \((\phi(u), \phi(v)) \in E_2\).

A type-restricted isomorphism as defined above may permute the inputs and outputs of a system. The results we derive in the remainder of this paper remain valid if we require a type-restricted isomorphism to leave the inputs and outputs in the same order. Thus, if desired, this condition may be added to Definition \([7]\).

In addition to the isomorphisms we have defined above, we will find it useful to consider isomorphisms on condensed graphs. The condensed graph corresponding to a given graph is the graph whose vertices are the components of the original graph. An edge from one component to another exists in the condensed graph if an edge from a vertex in one component to a vertex in

---

**Example 1.** Consider the system \( S = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \end{pmatrix} \). The associated graph \( G(S) \) of \( S \) is shown in Figure 1. Of the two functions \( \phi_1 \) and \( \phi_2 \) defined below, \( \phi_1 \) is type-restricted while \( \phi_2 \) is not.

\[
\phi_1(v) = \begin{cases} x_2(G(S)) & \text{if } v = x_1(G(S)) \\ x_1(G(S)) & \text{if } v = x_2(G(S)) \\ v & \text{otherwise} \end{cases}
\]

\[
\phi_2(v) = \begin{cases} u_1(G(S)) & \text{if } v = x_1(G(S)) \\ x_1(G(S)) & \text{if } v = u_1(G(S)) \\ v & \text{otherwise} \end{cases}
\]

The first kind of function we will consider is an isomorphism. A (directed)-graph isomorphism is defined in Definition \([4]\). Unless otherwise noted, we will additionally require that the isomorphism is type-restricted as defined above.

**Definition 7.** Let \( S_1 \) and \( S_2 \) be linear systems and let \( G(S_i) = (V_i, E_i) \) for \( i = 1, 2 \). A type-restricted function \( \phi : V_1 \to V_2 \) is a type-restricted isomorphism if:

1. \( \phi \) is a bijection, that is, it is both injective and surjective.
2. For any vertices \( u, v \in V_1 \), \((u, v) \in E_1\) if and only if \((\phi(u), \phi(v)) \in E_2\).

A type-restricted isomorphism as defined above may permute the inputs and outputs of a system. The results we derive in the remainder of this paper remain valid if we require a type-restricted isomorphism to leave the inputs and outputs in the same order. Thus, if desired, this condition may be added to Definition \([7]\).
another existed in the original graph. To formally state the definition of a condensed graph and a condensed-graph isomorphism, we begin by stating the definition of a component.

**Definition 8.** Let \( S \) be a linear system and let \( G(S) = (V, E) \) be its associated graph. For all \( a, b \in V \), let \( a \leftrightarrow b \) if and only if there exist directed paths from \( a \) to \( b \) and vice-versa in \( G(S) \). The relation \( \leftrightarrow \) defined above is an equivalence relation. The equivalence classes of this relation are called the strong components of \( G(S) \).

From the definition above, it is clear that the strong components of \( G(S) = (V, E) \) are sets of vertices that form a partition of \( V \). In order to define the condensed graph \( CG(S) \) of \( S \) and type-restricted isomorphisms on such graphs, we will need to assign a type to each component of \( G(S) \). To do this, we recall an observation previously made by Dion et al. in their survey of structured linear systems [2]. A formal proof of this observation is given in Appendix \( \Delta \).

**Observation 1.** Each input vertex \( u_i(G(S)) \) or output vertex \( y_j(G(S)) \) is the only element of its component in \( CG(S) \).

By the above observation, any component of \( G(S) \) with two or more elements must consist entirely of state variables. We will use this fact below to state the definition of a condensed graph and type-restricted mappings between such graphs.

**Definition 9.** Let \( S \) be a linear system with \( n_u \) inputs and \( n_y \) outputs and let \( G(S) = (V_G, E_G) \) be its associated graph. Furthermore, let \( c_1, c_2, \ldots, c_m \) be the strong components of \( G(S) \) with more than one element. Then, we define the vertex set \( V_{CG} \) by \( V_{CG} = \{u_1(G(S)), u_2(G(S)), \ldots, u_{n_u}(G(S)), c_1, c_2, \ldots, c_m, y_1(G(S)), y_2(G(S)), \ldots, y_{n_y}(G(S))\} \). The edge set \( E_{CG} \) contains the elements specified by the conditions below. The condensed graph \( CG(S) = (V_{CG}, E_{CG}) \).

1. For all \( 1 \leq i \leq n_u \) and \( 1 \leq j \leq m \), \( (u_i, c_j) \in E_{CG} \) if and only if a vertex \( v \in c_j \) exists such that \( (u_i, v) \in E_G \).
2. For all \( 1 \leq i, j \leq m \), \( (c_i, c_j) \in E_{CG} \) if and only if vertices \( v_1 \in c_i \) and \( v_2 \in c_j \) exist such that \( (v_1, v_2) \in E_G \).
3. For all \( 1 \leq i \leq m \) and \( 1 \leq j \leq n_y \), \( (c_i, y_j) \in E_{CG} \) if and only if a vertex \( v \in c_i \) exists such that \( (v, y_j) \in E_G \).
4. For all \( 1 \leq i \leq n_u \) and \( 1 \leq j \leq n_y \), \( (u_i, y_j) \in E_{CG} \) if and only if \( (u_i, v) \in E_G \).

In addition, we will denote the vertices \( u_i(G(S)) \) by \( u_i(CG(S)) \) and call these vertices input vertices. Similarly, the vertices \( y_j(G(S)) \) will be denoted by \( y_j(CG(S)) \) and will be called output vertices. Finally, we will call the vertices \( c_j \) state variable component vertices and denote them by \( c_j(CG(S)) \).

**Definition 10.** Let \( S_1 \) be a linear system with \( n_{u,1} \) inputs and \( n_{y,1} \) outputs and \( S_2 \) a linear system with \( n_{u,2} \) inputs and \( n_{y,2} \) outputs. Furthermore, let \( n_{c,i}, i = 1, 2 \), be the number of state variable component vertices of \( CG(S_i) \). Finally, let \( CG(S_i) = (V_i, E_i) \), for \( i = 1, 2 \). Then, a mapping \( \phi : V_1 \rightarrow V_2 \) is type-restricted if, for all \( v \in V_1 \), one of the following conditions holds for some \( i \) and \( j \):

1. \( 1 \leq i \leq n_{u,1} \wedge 1 \leq j \leq n_{u,2} \wedge v = u_i(CG(S)) \wedge \phi(v) = u_j(CG(S')) \)
2. \( 1 \leq i \leq n_{y,1} \wedge 1 \leq j \leq n_{y,2} \wedge v = y_i(CG(S)) \wedge \phi(v) = y_j(CG(S')) \)
3. \( 1 \leq i \leq n_{c,1} \wedge 1 \leq j \leq n_{c,2} \wedge v = c_i(CG(S)) \wedge \phi(v) = c_j(CG(S')) \)

Using the above definitions, we can define isomorphisms on condensed graphs, which we will refer to as CG-isomorphisms. As we define below, two systems \( S \) and \( S' \) such that a CG-isomorphism exists between their condensed graphs are called CG-isomorphic.

**Definition 11.** Let \( S \) and \( S' \) be linear systems. Then, a type-restricted isomorphism between \( CG(S) \) and \( CG(S') \) is said to be a condensed-graph (CG-) isomorphism between \( S \) and \( S' \). If a CG-isomorphism between \( S \) and \( S' \) exists, \( S \) and \( S' \) are said to be CG-isomorphic, written \( S \simeq_{CG} S' \).

### 3 Main results

#### 3.1 Graph isomorphism and its inadequacy

We begin by defining linear transformations of systems and considering how these transformations affect the associated graph of the system. Consider the system \( S = (A, B, C, D) \) with \( n_u \) inputs, \( n_s \) state variables and \( n_y \) outputs. The evolution over time of the state vector \( x_k \in \mathbb{R}^{n_s} \) and output \( y_k \in \mathbb{R}^{n_y} \) of \( S \), given an input \( u_k \in \mathbb{R}^{n_u} \), is given by (5) and (6). Let \( T \) be an invertible \( n_x \times n_x \) matrix and \( z_k = T x_k \). Then, the evolution over time of \( z_k \) and \( y_k \) is given by (7) and (8). We define the result of transforming the system \( S \) using the matrix \( T \) to be the system with state vector \( z_k \). This definition is stated formally below.

\[
z_{k+1} = T A z_k + T B u_k \quad (7)
\]
\[
y_k = C T z_k + D u_k \quad (8)
\]

**Definition 12.** The result of transforming the system \( S = (A, B, C, D) \) with \( n_s \) state variables using an invertible \( n_x \times n_x \) matrix \( T \), denoted \( T(S, T) \), is the system \((T A^{-1}, T B, C T^{-1}, D)\).
Our first result is that in many cases, we can find a new system with a non-isomorphic graph by diagonalizing the $A$-matrix of the system. Thus, it is possible for a linear transformation to result in a system with a different graph.

**Theorem 1.** For a system $S = (A,B,C,D)$, where $A$ is diagonalizable but not diagonal, there exists an invertible matrix $T$ such that $G(S)$ and $G(T(S,T))$ are non-isomorphic.

On the other hand, there do exist matrices $T$ such that $G(T(S,T))$ and $G(S)$ are isomorphic for all $S$. For any positive integer $n$, we call the set of all such $n \times n$ matrices $GI(n)$. The set $GI(n)$ is a subgroup of the group of invertible $n \times n$ matrices, as stated below.

**Definition 13.** The set $GI(n)$ consists of the $n \times n$ invertible matrices $T$ such that $G(S)$ and $G(T(S,T))$ are isomorphic for all linear systems $S$ with $n$ state variables.

**Theorem 2.** $GI(n)$ is a subgroup of all invertible $n \times n$ matrices. Equivalently, the following conditions hold:

1. $I_n \in GI(n)$
2. $GI(n)$ is closed under matrix multiplication
3. $GI(n)$ is closed under matrix inversion.

Though the above result shows that $GI(n)$ has some algebraic structure, it does not indicate how many and which elements this group has. Below, we state two results that show that certain classes of matrices are subsets of $GI(n)$. One of these classes, the permutation matrices, is defined below. In Theorem 14, $\{m_i\}$ denotes the diagonal matrix whose diagonal elements are $m_i$. Thus, $\{m_1, m_2, \cdots, m_n\}$ denotes the matrix $M$ given below.

$$M = \begin{bmatrix} m_1 & 0 & \cdots & 0 \\ 0 & m_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & m_n \end{bmatrix}$$

**Theorem 3.** For any diagonal matrix $M = \{m_i\}$, $1 \leq i \leq n$, such that $m_i \neq 0$ for all $i$, $M \in GI(n)$.

**Definition 14.** For a permutation $e_1, e_2, \cdots, e_n$ of the integers 1, 2, $\cdots$, $n$, the corresponding $n \times n$ permutation matrix is the matrix $P(e_1, e_2, \cdots, e_n) = (P_{ij})$, $1 \leq i, j \leq n$, where $P_{ij}$ is as given below.

$$P_{ij} = \begin{cases} 1 & \text{if } j = e_i \\ 0 & \text{otherwise} \end{cases}$$

**Theorem 4.** Any $n \times n$ permutation matrix is an element of $GI(n)$.

---

**Figure 2: The system $S_1$ from Example 2**

---

### 3.2 Graph homomorphism

In most of this paper, we concentrate on graph isomorphism. In principle, we could also have focused on a weaker relation, such as graph homomorphism. In this subsection, we state a number of results that indicate why we consider graph homomorphism to be inadequate for our purposes. We begin by defining a graph homomorphism below. Intuitively, a graph homomorphism is a mapping from one graph to another that conserves edges.

**Definition 15.** Let $S_1$ and $S_2$ be linear systems. Furthermore, let $G(S_i) = (V_i, E_i)$ for $i = 1, 2$. Then, a type-restricted function $\phi: V_1 \rightarrow V_2$ is a type-restricted homomorphism if for all $u, v \in V_1$, $(u, v) \in E_1$ implies $(\phi(u), \phi(v)) \in E_2$.

Our first example shows that the graph of every system is homomorphic to that of a standard system.

**Example 2.** Consider the system $S_1 = ([1], [1], [1], [1])$. The graph of $S_1$ is shown in Figure 2. Next, consider an arbitrary system $S = (A,B,C,D)$. The mapping $\phi$ defined below is a type-restricted homomorphism from $S$ to $S_1$.

$$\phi(v) = \begin{cases} u_1(G(S_1)) & \text{if } v = u_i(G(S)) \text{ for some } i \\ y_1(G(S_1)) & \text{if } v = y_i(G(S)) \text{ for some } i \\ x_1(G(S_1)) & \text{if } v = x_i(G(S)) \text{ for some } i \end{cases}$$

If graph homomorphism were an equivalence relation, the above example would be problematic since every system would have the same structure. Fortunately, graph homomorphism fails to be symmetric, and so cannot be an equivalence relation. This fact is shown in the following example.

**Example 3.** We will now show that, though a homomorphism from $S$ to $S_1$ exists for all $S$, the opposite is not necessarily the case. To do this, we consider the system $S_2 = ([0], [1], [1], [0])$, whose graph is shown in Figure 3. Since any homomorphism $\phi$ from $S_1$ to $S_2$ must be type-restricted, it follows that $\phi(x_1(G(S_1))) = x_1(G(S_2))$. But then, by the definition of a type-restricted homomorphism,


\( (x_1(G(S_2)), x_1(G(S_2))) \) should be an edge of \( G(S_2) \), since \( (x_1(G(S_1)), x_1(G(S_1))) \) is an edge of \( G(S_1) \). Since this is not the case, no type-restricted homomorphism from \( S_1 \) to \( S_2 \) can exist.

One way to make homomorphism an equivalence relation is to require not just that a homomorphism from \( G_1 \) to \( G_2 \) exists, but also that a homomorphism from \( G_2 \) to \( G_1 \) exists. Even with this interpretation, homomorphism can lead to strange results, as we show below.

**Example 4.** Our next example concerns the system \( S_3 = \left( \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \right) \). The mapping \( \phi \) below is a homomorphism from \( S_3 \) to \( S_2 \), and similarly, \( \psi \) is a homomorphism in the other direction.

\[
\phi(v) = \begin{cases} 
  u_1(G(S_2)) & \text{if } v = u_1(G(S_3)) \lor v = u_2(G(S_3)) \\
  y_1(G(S_2)) & \text{if } v = y_1(G(S_3)) \lor v = y_2(G(S_3)) \\
  x_1(G(S_2)) & \text{if } v = x_1(G(S_3)) \\
\end{cases}
\]

\[
\psi(v) = \begin{cases} 
  u_1(G(S_3)) & \text{if } v = u_1(G(S_2)) \\
  y_1(G(S_3)) & \text{if } v = y_1(G(S_2)) \\
  x_1(G(S_3)) & \text{if } v = x_1(G(S_2)) \\
\end{cases}
\]

So, if we consider two systems to have the same structure if homomorphisms exist between them in both directions, \( S_2 \) and \( S_3 \) have the same structure.

In the above example, we show that the systems \( S_2 \) and \( S_3 \) have the same structure. Intuitively, this should not be the case, since \( S_2 \) and \( S_3 \) have different numbers of inputs. From the above examples, we conclude that homomorphism, at least in the variants we have discussed here, is inadequate for our purposes since it will always lead to strange results.

### 3.3 Condensed-graph isomorphism

We will now consider how some of the results we derived earlier for graph isomorphism apply to condensed-graph isomorphism. Our first result is that the counterexample we presented for graph isomorphism also applies to condensed-graph isomorphism.

**Theorem 5.** A system \( S = (A, B, C, D) \) with a non-diagonal but diagonalizable, is not CG-isomorphic to \( T(S, T) \), where \( D = TAT^{-1} \) is diagonal.

In our next result, we show that there exists a class of systems whose condensed graphs contain 1 state variable component. Each of these systems has a diagonalizable \( A \)-matrix, allowing us to transform each system to another system where each state variable is in a component of its own.

**Theorem 6.** For each integer \( n \), there exists a system \( S = (A, B, C, D) \) with \( n \) state variables such that its condensed graph \( CG(S) \) contains 1 state variable component. In addition, the matrix \( A \) is diagonalizable but not diagonal, implying that there exists a system \( T(S, T) \) such that \( CG(T(S, T)) \) has \( n \) state variable components.

The results above indicate that the condensed-graph structure of a system is not conserved by linear transformations. Since linear transformations do not change the input/output behavior of a system, this implies that systems with different condensed-graph structures may produce the same input/output data. Therefore, we cannot identify the condensed-graph structure of a system using input/output data unless we have additional assumptions about the system. In other words, to determine the (condensed-)graph structure of a system, we need a parameterization of the set of linear systems we are interested in. Given a particular parameterization, we can ask when two systems in this parameterization have isomorphic structures. We will now consider this question for two parameterizations. The first parameterization concerns the minimal SISO systems with diagonal \( A \)-matrices. In the second parameterization, we consider minimal SISO systems whose \( A \)-matrices are in the natural normal forms given by Gantmacher [3].

For systems in the first parameterization, i.e. minimal SISO systems with diagonal \( A \)-matrices, we can characterize the existence of a type-restricted isomorphism between two systems. This result is stated below.

**Theorem 7.** The graphs of two minimal diagonal SISO realizations \( S = (A, B, C, D) \) and \( S' = (A', B', C', D') \) will be isomorphic if and only if the following conditions hold:

1. Either \( D = D' = 0 \) or both \( D \) and \( D' \) are non-zero.
2. \( A \) and \( A' \) have the same number of non-zero elements along their diagonals.
In the second parameterization, a system’s $A$-matrix is in one of the normal forms given by Gantmacher [3]. These normal forms are based on the elementary divisors and invariant polynomials of the matrix $A$ and are defined below. The concepts of invariant polynomials and elementary divisors are recalled in the Appendix.

**Definition 16.** Let $l(s) = s^n + a_1 s^{n-1} + a_2 s^{n-2} + \cdots + a_n$. The companion matrix $L$ for the polynomial $l(s)$ is the $n \times n$ square matrix shown below. As shown by Gantmacher [3, Ch. 6], $|\lambda I - L| = l(\lambda)$. Furthermore, all invariant polynomials $i_j$ of $L$ other than $i_1$ are equal to 1, and so $i_1 = |\lambda I - L|$.

$$ L = \begin{bmatrix} 0 & 0 & \cdots & 0 & -a_n \\ 1 & 0 & \cdots & 0 & -a_{n-1} \\ 0 & 1 & \cdots & 0 & -a_{n-2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 & -a_1 \end{bmatrix} $$

**Definition 17.** Let $A$ be a real matrix and $i_1, i_2, \ldots, i_r, i_{r+1}, \ldots, i_n$ be its invariant polynomials, such that the polynomials $i_1$ through $i_r$ are of positive degree and $i_1 = 1$ for $r+1 \leq j \leq n$. Then, the natural normal form of $A$ is the matrix $L = \{L_1, L_2, \ldots, L_r\}$ where $L_j$ is the companion matrix for the polynomial $i_j$.

**Definition 18.** Let $e_1, e_2, \ldots, e_k$ be the elementary divisors of $A$. The second natural normal form of $A$ is the matrix $L = \{L_1, L_2, \ldots, L_k\}$, where $L_i$ is the companion matrix of $e_i$.

**Remark 2.** As Gantmacher [3] shows, every matrix $A$ is similar to a matrix $A_1$ in first natural normal form and a matrix $A_2$ in second natural normal form. Thus, for every linear system $S = (A, B, C, D)$, there exist systems $S_1 = (A_1, B_1, C_1, D_1)$ and $S_2 = (A_2, B_2, C_2, D_2)$ such that $A_1$ is in first natural normal form and $A_2$ is in second natural normal form and both $S_1$ and $S_2$ are similar to $S$.

As we state below, if a SISO realization is minimal and its $A$-matrix is in first natural normal form, the matrix $A$ is a companion matrix. This implies that the system will be similar to the well-known canonical forms of such systems.

**Theorem 8.** If a SISO realization $(A, B, C, D)$ is minimal and the matrix $A$ is in first natural normal form, $A$ is a companion matrix.

Unlike first natural normal form, second natural normal form differs from the well-known canonical forms. In this form, the matrix $A$ is block-diagonal and each block corresponds to an elementary divisor of the matrix $A$. Below, we give a characterization of $CG$-isomorphism between two realizations whose $A$-matrices are in second natural normal form.

**Theorem 9.** Let $S_1 = (A_1, B_1, C_1, D_1)$ and $S_2 = (A_2, B_2, C_2, D_2)$ be two minimal SISO realizations where $A_1$ and $A_2$ are in second natural normal form such that 0 is not an eigenvalue of either $A_1$ or $A_2$. Then, $G(S_1)$ and $G(S_2)$ are $CG$-isomorphic if and only if:

1. The number of distinct irreducible polynomials that divide $|\lambda I - A_1|$ equals the number of distinct irreducible polynomials that divide $|\lambda I - A_2|$.
2. Either $D_1 = D_2 = 0$ or $D_1 \neq 0 \land D_2 \neq 0$.

### 3.4 Components of condensed graphs

In the previous section, we saw that the condensed-graph structure of a system is not uniquely determined by its input/output behavior. Thus, there may be a variety of systems $S$ and associated graphs $G(S)$ corresponding to any set of input/output data. In each of these graphs $G(S)$, some sets of state variables do not interact with any variables outside the set itself. Such sets correspond to isolated components in the graph $G(S)$. By permuting the state variables of $S$, we can ensure that the variables in each component form a sequence of consecutive variables, that is, they are the variables $x_1, x_{i+1}, \ldots, x_{i+j}$, for some $i$ and $j$. The resulting system will then have an $A$-matrix with a block-diagonal structure, where each isolated component in the graph corresponds to a diagonal block. Conversely, if $S$ has a block-diagonal $A$-matrix, each block of $A$ corresponds to an isolated component in $G(S)$. Thus, if we can find bounds on the number of diagonal blocks a block-diagonal realization similar to $S$ may have, the same bounds should apply to the number of isolated components in the graphs of these realizations. In this section, we derive bounds on the number of diagonal blocks the $A$-matrix of a system similar to a given system may have. To do so, we primarily consider the class of systems with block-diagonal $A$-matrices where each block is a companion matrix. However, the upper bound we derive applies to all block-diagonal realizations similar to the given system, no matter how their blocks are structured.

**Definition 19.** A system $S = (A, B, C, D)$ is said to be a block-companion realization if $A$ is block-diagonal and each diagonal block $A_i$ of $A$ is a companion matrix.

In our first result, we state that there exists a lower bound on the number of diagonal blocks of a block-companion realization similar to a given system. As we state in our second result, an upper bound on this number of blocks also exists.

**Theorem 10.** Let $S = (A, B, C, D)$ be a given linear system. Furthermore, let $\phi_i, 1 \leq i \leq m$ be the irreducible polynomials that divide $|\lambda I - A|$ and $k_i$ the number of
elementary divisors of $A$ of the form $\phi_i^l$. Then, every block-companion realization $S'$ similar to $S$ has at least $k = \max_k k_i$ diagonal blocks.

**Theorem 11.** Let $S = (A, B, C, D)$ be a given linear system and let $d$ be its number of elementary divisors. Then, any block-companion realization $S'$ similar to $S$ has at most $d$ diagonal blocks.

In the result below, we state that for any integer $l$ between the bounds we have previously indicated, we can find a block-companion realization similar to $S$ with $l$ diagonal blocks.

**Theorem 12.** Let $S = (A, B, C, D)$ be a given linear system and let $k$ and $d$ be the lower and upper bounds of Theorems 14 and 17 respectively. Then, for any integer $l$ in the interval $[k, d]$, there exists a block-companion realization similar to $S$ with $l$ diagonal blocks.

As noted above, our main result is stated in terms of block-companion realizations. If we consider block-diagonal realizations whose blocks have arbitrary shapes, we find that the upper bound given above still applies. This fact is stated as a remark below.

**Remark 3.** The upper bound of Theorem 11 also applies to arbitrary block-diagonal realizations, regardless of the shape of the blocks.

While the upper bound of Theorem 11 remains valid for arbitrary block-diagonal realizations, the lower bound does not. A counterexample is given below. This example also indicates that this lower bound does not apply even if we consider only minimal systems.

**Example 5.** First, we will construct a system whose elementary divisors are $\lambda - 1$, with multiplicity 3, and $\lambda - 2$, with multiplicity 1. To do so, we note that the matrix $A$ given below is in first natural normal form and has invariant polynomials $i_1 = (\lambda - 1)(\lambda - 2)$, $i_2 = i_3 = \lambda - 1$. Let $S = (A, I_4, I_4, 0)$. Since $B = C = I_4$, $S$ is both controllable and observable. Thus, $S$ is minimal, as required. Next, we consider $\mathbb{T}(S, T)$, with $T$ given below. As is readily verified, $\mathbb{T}(S, T) = (A_1, T, T^{-1}, 0)$, where $A_1$ is given below. We notice that $A_1$ has two diagonal blocks, even though the lower bound of Theorem 11 is three for $S$.

$$A = \begin{bmatrix} 0 & -2 & 0 & 0 \\ 1 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ -1 & 1 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

### 3.5 Structured systems and their graphs

In many applications, we can assume that some variables are not influenced by certain other variables. In other words, we can assume that a number of edges $(v_{i,1}, v_{i,2}), 1 \leq i \leq l$, do not occur in the graph of the system generating the input/output data under consideration. This assumption implies that the graph $G(S)$ of this system is a subgraph of the graph $G$ containing all the edges we have not excluded, i.e. the graph $G = (V, \{(v_{i,1}, v_{i,2}) \in V \times V \mid \exists 3 : v_{i,1} = v_{i,1} \wedge v_{i,2} = v_{i,2}\})$, where $V$ is the set of vertices of the graph $G(S)$. As we will see, many of the properties of systems in this class hold for almost all systems in the class. Furthermore, whether these properties hold for almost all systems can be determined by examining the graph $G$. We call the class of systems $S$ such that their graphs $G(S)$ are subgraphs of $G$ a structured linear system. Since the edges not present in $G$ are absent in $G(S)$ if and only if the corresponding entries in the system matrices of $S$ are zero, a system $S$ is a member of the structured linear system if and only if those entries are zero. Since the edges present in $G$ may or may not be present in $G(S)$, the other entries of the system matrices of $S$ are unconstrained, i.e. they are free parameters. We state a formal definition of a structured linear system below.

**Definition 20.** Let $T_{m \times n} = \{(i, j) \mid 1 \leq i \leq m \wedge 1 \leq j \leq n\}$ and $T(M) \subset T_{m \times n}$. The structured matrix $M$ corresponding to the subset $T(M)$ is the subset of $\mathbb{R}^{m \times n}$ given below.

$$M = \{M' \in \mathbb{R}^{m \times n} \mid \forall (i, j) \in T(M), M'_{ij} = 0\}$$

The set of all $m \times n$ structured matrices is denoted by $\mathbb{S}_{m \times n}^R$.

**Definition 21.** A structured linear system $S$ with $n_u$ inputs, $n_x$ state variables and $n_y$ outputs is defined as a 4-tuple $(A, B, C, D)^S$, where $A \in \mathbb{S}_{R \times n_x}^{n_u}$, $B \in \mathbb{S}_{R \times n_x}^{n_u}$, $C \in \mathbb{S}_{R \times n_y}^{n_u}$, and $D \in \mathbb{S}_{R \times n_y}^{n_u}$.
Connectivity structure of systems

Let $S = \{A', B', C', D'\} \in \mathcal{S}$ if $A' \in \mathcal{A}$, $B' \in \mathcal{B}$, $C' \in \mathcal{C}$ and $D' \in \mathcal{D}$. We say that $S'$ = \{(A', B', C', D')\} \in \mathcal{S}$ if $A' \in \mathcal{A}$, $B' \in \mathcal{B}$, $C' \in \mathcal{C}$ and $D' \in \mathcal{D}$. Then, the subset $\mathcal{S}' \subseteq \mathcal{S}$ is said to be generic. Formally, we say that a property $P$ of structured linear systems is generic if all parameter vectors for which this property holds are identical. In the parameterization, each entry in the system matrices that is not set to zero corresponds to a single variable.

Definition 22. Let $S \in \mathcal{S}^{m \times n}$ and let $T(M)$ and $S_{mn}$ be given as in Definition 20. Furthermore, let $\mathcal{T}(M) = T_{mn} \setminus T(M)$. Order the elements of $\mathcal{T}(M)$ according to the lexicographic ordering $(i, j) \leq (i', j') \iff i < i' \lor (i = i' \land j \leq j')$. Let $t_1, t_2, \ldots, t_k$ be the elements of $\mathcal{T}(M)$, enumerated in the above order. Then, the standard parameterization of the structured matrix $M$ is the function $f_M : \mathbb{R}^k \to \mathcal{S}$ defined by $f_M(v) = (e_{ij}(v))$, with $e_{ij}$ given below.

$$e_{ij}(v) = \begin{cases} 0 & \text{if } (i, j) \in T(M) \\ v_t & \text{if } (i, j) = t_t \end{cases}$$

The standard parameterization $f_M$ is a bijection between $\mathbb{R}^k$ and $\mathcal{S}$.

Definition 23. Let $S = (A, B, C, D)^S$ and let $f_A : \mathbb{R}^{k_A} \to A$ be the standard parameterization of $A$, with $f_B, k_B, f_C, k_C, f_D$ and $k_D$ defined similarly. Then, the standard parameterization of $S$ is the function $f_S : \mathbb{R}^{k_A + k_B + k_C + k_D} \to \mathcal{S}$ defined below.

$$f_S \left( \begin{bmatrix} x_A \\ x_B \\ x_C \\ x_D \end{bmatrix} \right) = (f_A(x_A), f_B(x_B), f_C(x_C), f_D(x_D))$$

The standard parameterization $f_S$ is a bijection between $\mathbb{R}^{k_A + k_B + k_C + k_D}$ and $\mathcal{S}$. We will denote $f_S(p)$ by $S_p$.

A useful property of structured linear systems is that many of their properties hold for almost all of systems within the class given by a certain structured linear system. Properties that hold for “almost all” systems are said to be generic. Formally, we say that a property $P$ is generic if all parameter vectors for which this property does not hold lie in the zero set of some polynomial. Equivalently, the set of parameter vectors for which $P$ does not hold is a subset of a proper variety as defined below.

Definition 24. Let $f$ be a polynomial function in $n$ indeterminates with real coefficients. Then, the subset $V_f = \{p \in \mathbb{R}^n | f(p) = 0\}$ is called the variety determined by $f$. If $V_f \neq \mathbb{R}^n$, $V_f$ is said to be a proper variety.

Remark 4. In the remainder of this paper, we will denote the variety determined by $f$ by $V_f$. Conversely, if we denote a variety by $V_f$, the polynomial determining this variety is denoted by $f$.

Definition 25. Let $f_S : \mathbb{R}^k \to \mathcal{S}$ be the standard parameterization of the structured linear system $S$. A property $P$ is said to be generic for $S$ if for all $p \in \mathbb{R}^k$ outside of a proper variety $V$, $P$ holds for $S_p$.

In our first result, we consider whether we can use input/output data to uniquely identify a particular member of a structured linear system. In other words, we consider whether knowing that certain edges do not occur in the system’s graph is sufficient to allow us to identify the system. We say that a system can be uniquely identified using input/output data if there does not exist a system with exactly the same input/output behavior. This is stated formally below.

Definition 26. $S = (A, B, C, D)$ and $S' = (A', B', C', D')$ are equivalent if, given zero initial conditions, the outputs $y_{S,k}$ of $S$ and $y_{S',k}$ of $S'$ are identical for all $k \geq 0$ for all input sequences $u_k$.

Definition 27. Let $S$ be a structured linear system. Then, $S$ is identifiable for $p$ if no parameter vector $q$ exists such that $p \neq q$ and $S_p$ and $S_q$ are equivalent.

In the result below, we state that unless the system matrix $C$ consists only of fixed zeroes, the structured linear system $S$ is not generically identifiable.

Theorem 13. Let $S = (A, B, C, D)^S$ be a structured linear system such that at least one entry of $C$ is not a fixed zero. Then, $S$ is not generically identifiable.

According to the result above, if a structured system $S = (A, B, C, D)^S$ is generically identifiable, all entries of the matrix $C$ are fixed zeroes. In other words, for any parameter value $p$, the output equation of the system $S_p$ reads $y_k = D u_k$. Therefore, a structured system that is generically identifiable can only represent very restricted input/output mappings.

Since all generically identifiable systems must have very limited modeling power, all structured systems that will occur in practical applications will not be generically identifiable. This implies that in real applications, the knowledge we have of the likely shape of the system's graph is insufficient to determine the system parameters exactly. In some cases, this knowledge may even be insufficient to uniquely determine the system’s structure. To guarantee that the system’s parameters can be found from input/output data, additional knowledge about the system is necessary.

In the parameterizations we discussed in Section 3.3, we assumed that the system had a particular shape and that it was minimal. Together, these assumptions allowed us to obtain characterizations of (CG-
isomorphism between systems. In our second application of structured linear systems, we show how the latter assumption, that is, that a system is minimal, implies that a system’s graph must satisfy a number of conditions. This result can be used with any particular parameterization to study the shape of the graph of a minimal system in this parameterization. In addition, this result shows that the assumptions we make about the properties a system has may affect its graph structure.

We begin by noting the following result, which states that a minimal system must be a member of some generically minimal structured linear system.

**Theorem 14.** A structured linear system $S$ is generically minimal if and only if there exists a minimal system $S_p \in S$.

This result implies that if a linear system $S$ is minimal in the usual sense, the structured linear system corresponding to its graph must be generically minimal. On the other hand, if the structured system linear system corresponding to the graph of $S$ is generically minimal, this does not guarantee that $S$ is minimal.

An ordinary linear system is minimal if and only if it is both controllable and observable. As we state below, the same holds generically for a structured linear system.

**Theorem 15.** A structured linear system $S$ is generically minimal if and only if it is both generically controllable and generically observable.

As given by Dion et al. [2], graph-theoretical conditions for the generic controllability of a structured linear system exist. Applying these conditions, we find the graph-theoretical characterization of generic minimality stated below in Theorem 15. The graph of a structured linear system, to which this characterization applies, is defined formally below. Intuitively, an edge exists in this graph if and only if the corresponding entry in the system matrices is not a fixed zero.

**Definition 28.** The graph of a structured linear system $S$ is the graph of the system $S_v$, where $v$ is the vector such that $v_i = 1$ for all $i$.

Below, we define the concepts used in the statement of Theorem 15. Similar definitions of these concepts are given by Dion et al. [2].

**Definition 29.** Let $S = (A, B, C, D)^S$ be a structured linear system. A finite sequence of vertices $v_1 v_2 \cdots v_k$ in $G(S)$ such that no two vertices $v_i$ and $v_j$ are equal except possibly $v_1$ and $v_k$ is called a path of length $k$. If $v_1 = v_k$, the path is called a cycle. Otherwise, the path is a simple path. Furthermore, if $v_1 = u_j(G(S))$ for some $i$, the path is called $U$-rooted. Similarly, if $v_k = y_j(G(S))$ for some $j$, the path is said to be $Y$-topped.

**Definition 30.** A path $u_1 u_2 \cdots u_k$ is said to cover the vertices $u_1$ through $u_k$. Two paths $p = u_1 u_2 \cdots u_k$ and $q = u'_1 u'_2 \cdots u'_l$ are said to be disjoint if no vertex $w$ exists such that both $p$ and $q$ cover $w$.

A set of simple paths such that any two of them are disjoint is called a path family. If every element of the family is $U$-rooted, the family is said to be $U$-rooted. Similarly, a path family of which every member is $Y$-topped is called $Y$-topped. A set of cycles such that any two of them are disjoint is called a cycle family.

The union of two path or cycle families $F_1$ and $F_2$ is said to be disjoint if for all $p_1 \in F_1$ and $p_2 \in F_2$, $p_1$ and $p_2$ are disjoint.

**Theorem 16.** A structured linear system $S$ is generically minimal if and only if its graph $G(S)$ satisfies the following conditions.

1. Every vertex $x_i(G(S))$ is the end vertex of a $U$-rooted path in $G(S)$
2. There exists a disjoint union of a $U$-rooted path family and a cycle family in $G(S)$ that covers all the vertices $x_i(G(S))$
3. Every vertex $x_i(G(S))$ is the first vertex of a $Y$-topped path in $G(S)$
4. There exists a disjoint union of a $Y$-topped path family and a cycle family in $G(S)$ that covers all the vertices $x_i(G(S))$.

As we have said previously, we can use the graph-theoretical characterization of generic minimality given by the previous result to determine if a given linear system may be minimal. This result is stated formally below.

**Theorem 17.** If a linear system $S$ is minimal, its graph must satisfy the conditions of Theorem 16.

4 Proofs of the main results

4.1 Graph isomorphism and its inadequacy

We begin by stating the proof of Theorem 1.

**Proof of Theorem 1.** Since $A$ is diagonalizable, there exists an invertible matrix $T$ such that $A = T M T^{-1}$, where $M$ is a diagonal matrix. Equivalently, $M = T^{-1} A T$. This implies that $T (S, T^{-1}) = (M, B', C', D')$, for some $B'$, $C'$ and $D'$. Since $M$ is diagonal, no edge of the form $(x_i(G(T(S, T^{-1}))), x_j(G(T(S, T^{-1}))))$ exists in $G(T(S, T^{-1}))$, for distinct $i$ and $j$. However, since $A$ was not diagonal, at least one edge of this form exists in $G(S)$. An isomorphism must preserve this edge, which is impossible. Therefore, $G(S)$ and $G(T(S, T^{-1}))$ are non-isomorphic, as claimed. □
In order to prove Theorem 2, we first take a closer look at the type-restricted isomorphism relation. We formally re-define this relation below.

Definition 31. The systems $S$ and $S'$ are said to be type-restricted isomorphic, denoted $S \simeq S'$, if a type-restricted isomorphism exists between $G(S)$ and $G(S')$.

As we state below, the relation defined above is an equivalence relation. The proof of this lemma is deferred to an appendix.

Lemma 1. Type-restricted isomorphism (written $S \simeq S'$), is an equivalence relation, that is:

1. $S \simeq S$, for all $S$
2. If $S \simeq S'$, then $S' \simeq S$
3. If $S \simeq S'$ and $S' \simeq S''$, then $S \simeq S''$

Using Lemma 1 we can now state the proof of Theorem 2.

Proof of Theorem 2. To prove the first condition, notice that $T(S, I_n) = S$ for all $S$. Therefore, $T(S, I_n) \simeq S$.

Next, let $T, T' \in GI(n)$ and let $S$ be some realization. Then, $T(T(S, T), T') \simeq (S, T)$, for any definition of $GI(n)$. For the same reason, $T(T(S, T), T') \simeq (S, T)$, and so $T(T(S, T), T') \simeq S$. Noticing that $T(T(S, T), T') = T(S, T')$, we find that $T \in GI(n)$, as required.

Finally, let $T \in GI(n)$. Then, let $S' = T(S, T')$, for some $S$. It follows that $S = T(S', T)$. Therefore, $S \simeq S'$, by the definition of $GI(n)$. Thus, $S' \simeq S$ and so $T' \in GI(n)$.

Next, we state the proof of Theorem 2.

Proof of Theorem 2. Let $M$ be given and $S = (A, B, C, D)$ be an arbitrary realization. Then, $T(S, M) = (L_{M_{ij}}^{-1})$. Notice that if $L = (L_{ij})$, then $ML = (m_{ij})$. Similarly, $LM^{-1} = (L_{ij}^{-1})$. Therefore, $T(S, M) = (m_{ij})$. Since both $m_{ij}$ and $m_{ij}^{-1}$ are clearly non-zero for all $i, j$, each entry of each matrix in $T(S, M)$ is non-zero if and only if the corresponding entry in $S$ is non-zero. Therefore, the graph of $S$ is the same as that of $T(S, M)$, and so $S \simeq T(S, M)$. Since $S$ was arbitrary, $M \in GI(n)$, as claimed.

To prove that permutation matrices are elements of $GI(n)$, we recall a number of properties of such matrices below. The first of these properties is proved in our appendix. The second follows straightforwardly from matrix multiplication.

Lemma 2. A permutation matrix $P(e_1, e_2, \ldots, e_n)$ is orthogonal.

Lemma 3. For a matrix $A = (A_{ij})$ and a permutation matrix $P(e_1, e_2, \ldots, e_n)$, $P(e_1, e_2, \ldots, e_n)A = (A_{ij})$ and $AP(e_1, e_2, \ldots, e_n) = (A_{ij})$, whenever these products exist.

Below, we show that permutation matrices are indeed elements of $GI(n)$, proving Theorem 3.

Proof of Theorem 3. Let $S = ((A_{ij}), (B_{ij}), (C_{ij}), (D_{ij}))$ and let $P = P(e_1, e_2, \ldots, e_n)$ be an arbitrary permutation matrix. By Lemma 2, $P(B_{ij}) = (B_{ij})$ and $(C_{ij})P^T = (C_{ij})$. Similarly, $P(A_{ij}) = (A_{ij})$, and so $(P(A_{ij})P^T = (A_{ij})$. Thus, $T(S, P) = ((A_{ij}), (B_{ij}), (C_{ij}), D)$. We claim that $\phi$ defined below is a type-restricted isomorphism between $S$ and $T(S, P)$.

From the definition of $\phi$, it is clear that $\phi$ is type-restricted. Since the integers $e_1, e_2, \ldots, e_n$ are a permutation of the integers from 1 through $n$, it is also clear that $\phi$ is a bijection. Thus, we only need to verify that $\phi$ preserves edges. To do so, we consider the four types of edges given in Corollary 1.

First, we have edges of the form $(x_{e_j}, x_{e_i})$. Then, $\phi(x_{e_j}) = x_j$ and $\phi(x_{e_i}) = x_i$. The edge $(x_j, x_i)$ exists in $G(T(S, P))$ if and only if $(PA_{ij}) = (A_{ij})$. Thus, this edge exists if and only if $(x_{e_j}, x_{e_i})$ is an edge of $G(S)$.

Second, consider edges of the form $(u_i, x_{e_j})$, which exist in $G(S)$ if and only if $B_{e_j} \neq 0$. By definition, $(\phi(u_i), \phi(x_{e_j})) = (u_i, x_{e_j})$. This edge exists in $G(T(S, P))$ if and only if $(PB_{ij}) = (B_{e_j})$. Thus, this edge is conserved.

Third, consider an edge of the form $(x_{e_j}, y_j)$, which is an edge of $G(S)$ if and only if $C_{ij} \neq 0$. Using $\phi$ again, $(\phi(x_{e_j}), \phi(y_j)) = (x_{e_j}, y_j)$. This edge exists in $G(T(S, P))$ if and only if $(CP^T_{ij}) = (C_{ij})$. Thus, edges of this form are also conserved.

Finally, all edges of the form $(u_i, y_j)$ correspond to edges $(u_i, y_j)$, since outputs and inputs are not permuted by $\phi$. Since both $S$ and $T(S, P)$ have the same matrix $D$, these edges are clearly conserved. Thus, $\phi$ is a type-restricted isomorphism. Since $S$ was arbitrary, $P \in GI(n)$.

4.2 Condensed-graph isomorphism

Below, we formally prove Theorem 4.

Proof of Theorem 4. First, we note that since $D$ is diagonal, $G(T(S, T))$ has no components with more than one element. Thus, if $G(S)$ has any components with
more than one element, $CG(S)$ and $CG(T(S,T))$ are of different order and so cannot be isomorphic. Second, if $G(S)$ has no such components, $CG(S)$ and $CG(T(S,T))$ are identical to $G(S)$ and $G(T(S,T))$ respectively. We have shown previously that no type-restricted isomorphism between these graphs can exist.

We will now prove Theorem 6 in two steps. In the first step, we construct a transfer function such that the realization in observable canonical form of this transfer function has 1 state variable component in its condensed graph. Then, we show that we can construct a transfer function of this type where the resulting realization has a diagonalizable A-matrix.

**Lemma 4.** Let $H(s) = \frac{b_0 + b_1 s + \cdots + b_n s^n}{\lambda^n + a_1 \lambda^{n-1} + \cdots + a_n}$ be a given transfer function. Then, the condensed-graph of the observable canonical form of $H(s)$ contains exactly one state-variable component if $a_n \neq 0$.

**Proof.** Consider a transfer function $H(s) = \frac{b_0 + b_1 s + \cdots + b_n s^n}{\lambda^n + a_1 \lambda^{n-1} + \cdots + a_n}$. The observable canonical form of $H(s)$, as described by Vaccaro [8, Ch. 3.3.2], is given by $S = (A, B, C, D)$, where $A$, $B$, $C$, and $D$ are given below. We notice that for all values of the coefficients $a_i$, the edges $(x_i, x_{i-1})$ for $2 \leq i \leq n$ exist in $G(S)$. If $a_n \neq 0$, the edge $(x_1, x_n)$ will also exist in $G(S)$, completing a cycle containing all state variables. Thus, as claimed, $G(S)$ will then have a single component containing all state variables.

$$A = \begin{bmatrix} -a_1 & 1 & 0 & 0 & \cdots & 0 \\ -a_2 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{n-1} & 0 & 0 & 0 & \cdots & 1 \\ -a_n & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad B = \begin{bmatrix} b_1 - a_1 b_0 \\ b_2 - a_2 b_0 \\ \vdots \\ b_n - a_n b_0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}, \quad D = [b_0]$$

To complete the proof of Theorem 6, we construct a transfer function satisfying the conditions of the previous lemma such that the observable canonical form has a diagonalizable A-matrix. This construction is given below.

**Proof of Theorem 6.** Let $H(s) = \frac{1}{\prod_{i=1}^{n}(s - \lambda_i)}$. Since $(0-i)$ is non-zero for any $1 \leq i \leq n$, the coefficient $a_n$ of the denominator of $H(s)$ is non-zero. Let $S$ be the observable canonical form of $H(s)$. As we have shown previously, $G(S)$ contains a component containing all state variables.

Let $A$ be the A-matrix of $S$. Since the eigenvalues of $A$ are the poles of $H(s)$, these eigenvalues are the integers from 1 through $n$. As $A$ has $n$ distinct eigenvalues, $A$ is diagonalizable. Thus, we can find a matrix $T$ such that $G(T(S,T))$ has $n$ components containing state variables.

In order to prove our isomorphism results, we introduce a number of concepts that we have found useful. Both of these concepts are sets of state variables. We recall from Definition 4 that the state variables in the graph $G(S)$ of some system $S$ are the vertices $x_i(G(S))$. The first concept is that of a trap, i.e. a set of state variables with no edges to any vertex not in the set itself. The second is the similar concept of an unreachable set, i.e. a set of state variables such that no vertex outside of the set itself has an edge to any of the variables in the set. These concepts are defined formally below.

**Definition 32.** A non-empty set of state variables $S$ in the graph $G$ of some system is called a trap if there does not exist a vertex $x \in S$ and a vertex $y \notin S$ such that $(x, y)$ is an edge of $G$.

**Definition 33.** A non-empty set of state variables $S$ in the graph $G$ of some system is called an unreachable set if there does not exist a vertex $x \in S$ and a vertex $y \notin S$ such that $(y, x)$ is an edge of $G$.

A useful property of traps and unreachable sets is that any system whose graph contains either an unreachable set or a trap is non-minimal. We state this below in two lemmas. The proof of the first of these is in an appendix, the second has a proof very similar to that of the first.

**Lemma 5.** A realization $S = (A, B, C, D)$ whose graph contains a trap is not observable, that is, its observability matrix $O = \begin{bmatrix} C & CA & \cdots & CA^{n-1} \end{bmatrix}$ has rank less than $n$.

**Lemma 6.** A realization $(A, B, C, D)$ whose graph contains an unreachable set is uncontrollable, that is, the rank of its controllability matrix $C = \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix}$ is less than $n$.

**Remark 5.** A result similar to Lemma 6 is described by Dion et al. [2]. Though we will not formally prove this here, the presence of an unreachable set in a system’s graph is equivalent to the system’s being in Form I as given by Dion et al. [2].

We now note that if, in $S = (A, B, C, D)$, the matrix $A$ is diagonal, no edges between distinct state variables exist. Thus, each state variable is a trap if it has no edge to the output $y$, and an unreachable set if it has no edge from the input. We state this formally below.

**Observation 2.** In the graph of a minimal diagonal realization, each state variable has an edge from the input and an edge to the output.

(v. September 20, 2011, p.14)
We now note that an isomorphism between diagonal realizations will preserve edges involving a state variable \( x_i \) if and only if it preserves edges of the form \((x_i, x_i)\). This is stated formally below.

**Observation 3.** An isomorphism \( \phi \) between the graphs \( G = (V, E) \) and \( G' = (V', E') \) of two diagonal realizations may have \( \phi(x_i) = x_j \) if and only if either \((x_i, x_i) \in E \) and \((x_j, x_j) \in E' \) or \((x_i, x_j) \notin E \) and \((x_j, x_j) \notin E' \).

Using the above observations, we will now prove Theorem\( \text{7} \).

**Proof of Theorem\( \text{7} \)** To prove necessity, let \( G(S) \) and \( G(S') \) be isomorphic. Then, since \((u, y)\) is an edge of either both \( G(S) \) and \( G(S') \) or of neither of these graphs, either both \( D \) and \( D' \) are non-zero or \( D = D' = 0 \). Since any isomorphism between \( G(S) \) and \( G(S') \) must conserve edges of the form \((x_i, x_i)\), \( G(S) \) and \( G(S') \) have equal numbers of vertices for which such an edge exists. Since each such vertex corresponds to a non-zero diagonal entry of the matrix \( A \) or \( A' \), the numbers of such entries are equal.

To prove sufficiency, we construct a bijection \( \phi \). Let \( x_{11}, x_{12}, \ldots, x_{1m_1} \) be the state variable vertices in \( G(S) \) for which an edge of the form \((x_i, x_i)\) exists, that is, the state variable vertices corresponding to non-zero diagonal entries of \( A \). Let \( x_{21}, x_{22}, \ldots, x_{2m_2} \) be the other state variable vertices of \( G(S) \). Define \( x'_{11}, x'_{12}, \ldots, x'_{1m_1} \) and \( x'_{21}, x'_{22}, \ldots, x'_{2m_2} \) similarly for \( G(S') \). Then, we claim that the function \( \phi \) defined below is an isomorphism between \( G(S) \) and \( G(S') \).

\[
\phi(v) = \begin{cases} 
  u_1(G(S')) & \text{if } v = u_1(G(S)) \\
  y_1(G(S')) & \text{if } v = y_1(G(S)) \\
  x'_{1i} & \text{if } v = x_{1i} \\
  x'_{2i} & \text{if } v = x_{2i} 
\end{cases}
\]

\( \phi \) is clearly a bijection. It is also clear from the definition of \( \phi \) that \( \phi \) is type-restricted. Thus, we only need to verify that \( \phi \) preserves edges. Since \( S \) is minimal, we know that for any state variable \( x_i, (u, x_i) \) and \((x_i, y)\) are edges of \( G(S) \). For the same reason, we know the same holds for \( \phi(x_i) \). Since both \( A \) and \( A' \) are diagonal, no edges between distinct state variables exist in either \( G(S) \) or \( G(S') \). By the definition of \( \phi \), it is clear that \( \phi \) also conserves edges of the form \((x_i, x_i)\). Finally, since \( D \) and \( D' \) are either both zero or both non-zero, \((u, y)\) is either an edge of both \( G(S) \) and \( G(S') \) or of neither of these graphs.

To prove Theorem\( \text{8} \) we will use the following lemma. The proof of this result is deferred to an appendix.

**Lemma 7.** If, for a SISO realization \((A, B, C, D)\), the minimal polynomial \( i_1 \) of \( A \) is not equal to \(|\lambda I - A|\) the realization is non-minimal.

Using the above lemma, we state the proof of Theorem\( \text{8} \) below.

**Proof of Theorem\( \text{8} \)** Using the above lemma, we see that if \( S = (A, B, C, D) \) is a minimal SISO realization, the minimal polynomial \( i_1 \) of \( A \) coincides with \(|\lambda I - A|\). Since the product of all minimal polynomials \( i_j \) of \( A \) also coincides with \(|\lambda I - A|\), \( i_j = 1 \) for \( j > 1 \). Thus, the first natural form of \( A \) consists of a single diagonal block that is the companion matrix of \( i_1 = |\lambda I - A| \). Therefore, we have established Theorem\( \text{8} \).

To prove our isomorphism result for realizations with \( A \)-matrices in second natural form, we begin by considering the graph of each block of the second natural form. As stated below, these graphs either have Hamiltonian cycles or are directed paths.

**Lemma 8.** For an irreducible polynomial \( \phi(\lambda) \), the graph of the companion matrix for polynomials of the form \( \phi(\lambda)^k \), where \( k > 0 \), satisfies one of the following conditions:

1. \( \phi(0) \neq 0 \) and the graph is Hamiltonian
2. \( \phi(0) = 0 \) and the graph is a directed path.

Since the graph of each block of \( A \) is a subgraph of the graph of \( S = (A, B, C, D) \), we have the following corollary of Lemma\( \text{8} \).

**Corollary 2.** If \( S = (A, B, C, D) \), where \( A \) is in second natural normal form, each block of \( A \) corresponds to either a Hamiltonian component or a directed path in \( G(S) \).

By applying the ideas of traps and unreachable sets to realizations in second natural normal form, we find the following lemma.

**Lemma 9.** If \( S = (A, B, C, D) \) is a minimal SISO realization and \( A \) is in second natural normal form, each component consisting of state variables in \( G(S) \) satisfies one of the following two conditions. Furthermore, these conditions are sufficient to guarantee that \( G(S) \) contains neither traps nor unreachable sets.

1. The component is Hamiltonian and state variables \( x_i \) and \( x_j \) in the component exist such that \((u, x_i)\) and \((x_j, y)\) are both edges of \( G(S) \)
2. The component is a directed path consisting of state variables \( x_{i_1}, x_{i_1+1}, \ldots, x_{i_m} \) and both \((u, x_{i_1})\) and \((x_{i_m}, y)\) are edges of \( G(S) \).

Additionally, we obtain the below lemma and its corollary by applying Lemma\( \text{7} \) to realizations in second natural normal form.

**Lemma 10.** If \( S = (A, B, C, D) \) is a SISO realization, where \( A \) has two elementary divisors of the form \( \phi(\lambda)^k \), for some irreducible \( \phi(\lambda) \), \( S \) is non-minimal.

(v. September 20, 2011, p.15)
Corollary 3. If \( S = (A, B, C, D) \) is a minimal SISO realization and \( A \) is in second natural normal form, the number of blocks in \( A \) equals the number of distinct irreducible polynomials that divide \(|\lambda - A|\).

Using the above lemmas, we now state the proof of Theorem 9.

Proof of Theorem 9. To prove necessity, suppose \( G(S_1) \) and \( G(S_2) \) are CG-isomorphic. Then, these graphs must have the same number of state-variable components. Since 0 is not an eigenvalue of either \( A_1 \) or \( A_2 \), each block in \( A_1 \) and \( A_2 \) corresponds to a Hamiltonian component, by Lemma 8. Thus, the number of such blocks must be equal. The first condition then follows from our corollary above. To show the second condition, notice that the edge \((u, y)\) in \( G(S_1) \) must exist if and only if the same edge exists in \( G(S_2) \). The second condition is obtained by restating this in terms of \( D_1 \) and \( D_2 \).

To prove sufficiency, suppose the conditions are satisfied. Then, let \( c_1, c_2, \ldots, c_m \) and \( c'_1, c'_2, \ldots, c'_m \) be the state-variable components of \( G(S_1) \) and \( G(S_2) \) respectively. By our corollary above, \( A_1 \) and \( A_2 \) have an equal number of diagonal blocks. As used above, this implies that \( m = m' \). We will now show that \( \phi \), as defined below, is a CG-isomorphism between \( G(S_1) \) and \( G(S_2) \).

\[
\phi(c) = \begin{cases} 
u_1(G(S_2)) & \text{if } c = \nu_1(G(S_1)) \\ 
u_1(G(S_2)) & \text{if } c = \nu_1(G(S_1)) \\ c'_1 & \text{if } c = c'_1 \end{cases}
\]

Clearly, \( \phi \) is type-restricted and, since \( m = m' \), \( \phi \) is a bijection. Thus, we verify that \( \phi \) conserves edges. Since each diagonal block in both \( A_1 \) and \( A_2 \) is a companion matrix corresponding to a polynomial with no root equal to zero, each component is either non-trivial and Hamiltonian or trivial and corresponding to a non-zero \((1 \times 1)\) matrix. Therefore, \((c_i, c_j)\) is an edge of \( G(S_1) \) for all \( i \), and similarly for \((c'_i, c'_j)\) in \( G(S_2) \). Since each component corresponds to a diagonal block, no edge of the form \((c_i, c_j)\) for distinct \( i \) and \( j \) exists in \( G(S_1) \) and similarly in \( G(S_2) \). Furthermore, since both realizations are minimal, \((u, c_i)\) and \((c_i, y)\) are edges of \( G(S_1) \) and similarly in \( G(S_2) \). Finally, \((u, y)\) is an edge of \( G(S_1) \) if and only if \( D_1 \neq 0 \) and similarly for \( G(S_2) \) and \( D_2 \). Since \( D_1 \) and \( D_2 \) are either both zero or both non-zero, \( \phi \) preserves this edge.

4.3 Components of condensed graphs

To prove our main results in this subsection, we will need the following theorem from Gantmacher [4]. This result allows us to find the elementary divisors of a block-diagonal matrix using those of the diagonal blocks.

**Theorem 18.** Let \( A = \{A_i\} \). Then, \( A \) has all the elementary divisors of the \( A_i \), and no others.

We begin by proving Theorem 10 i.e. that there exists a certain lower bound on the number of diagonal blocks in a block-companion realization similar to a given system. To do so, we recall from our definition of a companion matrix that each companion matrix \( L \) has only one invariant polynomial not equal to 1. This implies that for any irreducible polynomial \( \phi \) that divides \(|\lambda - L|\), \( L \) has exactly one elementary divisor of the form \( \phi^k \).

Proof of Theorem 11. Let \( S' = (A', B', C', D') \) be an arbitrary block-diagonal realization consisting of companion blocks that is similar to \( S \). Each block of \( A' \) is a companion matrix and so cannot have two elementary divisors corresponding to the same irreducible polynomial. Since \( A' \) is similar to \( A \), \( A' \) has the same elementary divisors as \( A \). Thus, there exists an integer \( i \) such that \( k_i = k \) and \( A' \) has \( k_i \) elementary divisors of the form \( \phi_i^k \). Since each block of \( A' \) has at most 1 elementary divisor of this form, \( A' \) must have at least \( k \) such blocks.

In a block-companion realization, each block of the matrix \( A \) corresponds to some nonzero number of elementary divisors of \( A \). Thus, the maximum number of diagonal blocks is the number of elementary divisors of \( A \). This argument establishes the upper bound of Theorem 11. In our discussion above, we did not use the assumption that the diagonal blocks of \( A \) were companion matrices. Thus, we have also proven Remark 3.

The proof of the remaining result, Theorem 12 is split into two parts. First, we show that the bounds given by the previous two results are sharp. In other words, there exist realizations with a number of blocks equal to the lower bound and realizations with a number of blocks equal to the upper bound. These results are stated below as lemmas. The proofs of these lemmas are given in Appendix A.5.

**Lemma 11.** Let \( S = (A, B, C, D) \) and \( k \) be the bound of Theorem 11. Then, there exists a block-companion realization \( S' \) similar to \( S \) such that \( S' \) has \( k \) diagonal blocks.

**Lemma 12.** Let \( A \) have \( n \) elementary divisors. Then, there exists a block-companion realization similar to \( S = (A, B, C, D) \) with \( n \) diagonal blocks.

In the following lemma, we state that if we have a block-companion realization with \( l \) blocks, we can rearrange the elementary divisors of these blocks to find a block-companion realization with \( l + 1 \) blocks, provided

---

3Recall that \( \{A_i\} \) denotes the block-diagonal matrix consisting of the blocks \( A_i \)

(v. September 20, 2011, p.16)
there exists a block-companion realization similar to $S$. Thus, $\tilde{V} \not\in (S)$.

Lemma 14. Let $S = (A, B, C, D)$ be a block-companion realization such that $A$ has $n$ elementary divisors and $l$ diagonal blocks. Then, either $l = n$ or there exists a block-companion realization similar to $S$ with $l + 1$ diagonal blocks.

Proof of the remainder of Theorem 12. By Lemma 11, there exists a block-companion realization similar to $S$ with $k$ diagonal blocks. By applying Lemma 13 $l - k$ times, we can find a sequence of realizations $S_1, S_2, \ldots, S_{l-k}$ such that $S_l$ is a block-companion realization similar to $S$ and $S_i$ has $k + i$ diagonal blocks. Thus, $S_{l-k}$ is the required realization.

4.4 Structured systems and their graphs

The first result we will prove is Theorem 13. To do so, we begin by stating the fact that $T(S, M)$ and $S$ have the same input/output relation in terms of our definition of equivalence. This fact is well-known and is stated in many textbooks on linear systems, such as the book by Vaccaro [8, Section 3.3.4]. We reproduce a formal proof of this result in the appendix.

Lemma 15. Let $S = (A, B, C, D)$ and $T(S, M)$ be equivalent for all $M$.

The next step of our proof is to show that a structured linear system satisfying the condition of Theorem 13 is generically not identifiable. We note that this result is subtly different from our theorem, which requires us to establish that this kind of structured linear system is not generically identifiable. Still, the below lemma constitutes a major part of our proof. Hence, its proof is stated here in full.

Lemma 16. Let $S$ be a block companion realization such that for every parameter vector $p$, $p \in V_f$ if and only if $S_p$ is not minimal.

Proof. Let $x$ be a vector consisting of $n$ indeterminates $x_1, x_2, \ldots, x_n$. Then, $S_x$ is a system whose system matrices are polynomial matrices in the variables $x_i$. Let $O$ be the observability matrix of $S_x$ and $C$ its controllability matrix. Clearly, $O$ and $C$ are polynomial matrices in the variables $x_i$. Let $f_o$ be the sum of the squares of all maximal-order minors of $O$, and similarly for $f_c$ using minors of $C$. Thus, for any parameter vector $p$, $f_o(p) = 0$ if and only if $S_p$ is not observable, and $f_c(p) = 0$ if and only if $S_p$ is not controllable. It follows that $f_o(p)f_c(p)$ is zero if and only if $S_p$ is not minimal. Thus, the variety $V_{f_o f_c}$ is the variety we require.

The proof of Theorem 14 using the above lemma is stated below.

Proof of Theorem 14. By Lemma 17, there exists a variety $V_f$ such that for every parameter vector $p$, $p \in V_f$ if and only if $S_p$ is not minimal. Thus, $V_f$ is proper if and only if there exists a parameter vector $p$ such that $S_p$ is minimal. Therefore, if there exists a parameter vector $p$ such that $S_p$ is minimal, $V_f$ is a proper variety containing the vectors $q$ for which $S_q$ is not minimal. Thus, $S$ is then generically minimal. If $S$ is generically minimal, $V_f$ must be proper, and so there exists a vector $p \not\in V_f$ such that $S_p$ is minimal.

Below, we state the proof of Theorem 15.

Proof of Theorem 15. To prove necessity, assume $S$ is generically minimal. Then, there exists a proper variety $V_f$ such that if $S_p$ is not minimal, the parameter vector $p \in V_f$. Let $q$ be a parameter vector such that

To complete the proof, we will need an additional lemma. This lemma states that if a property $P$, such as being not identifiable, holds generically, then its complement, such as being identifiable, does not hold generically. The proof of this result is deferred to an appendix since it uses some technical concepts.

Lemma 16. Let $P$ be generic for $S$. Then, $\neg P$ is not generic for $S$.

The above lemmas are sufficient to prove our result. Hence, we state the proof of Theorem 13 below.

Proof of Theorem 13. By Lemma 16, $S$ is generically not identifiable. The result then follows from Lemma 16.

To prove Theorem 14, we will use the following Lemma, which states that the parameter vectors for which a structured linear system is not minimal are always elements of some variety. Since this lemma is nearly sufficient to prove our result, we will state its proof in full.

Lemma 17. Let $S$ be a structured linear system. Then, there exists a variety $V_f$ such that for every parameter vector $p$, $p \in V_f$ if and only if $S_p$ is not minimal.

Proof. Let $x$ be a vector consisting of $n$ indeterminates $x_1, x_2, \ldots, x_n$. Then, $S_x$ is a system whose system matrices are polynomial matrices in the variables $x_i$. Let $O$ be the observability matrix of $S_x$ and $C$ its controllability matrix. Clearly, $O$ and $C$ are polynomial matrices in the variables $x_i$. Let $f_o$ be the sum of the squares of all maximal-order minors of $O$, and similarly for $f_c$ using minors of $C$. Thus, for any parameter vector $p$, $f_o(p) = 0$ if and only if $S_p$ is not observable, and $f_c(p) = 0$ if and only if $S_p$ is not controllable. It follows that $f_o(p)f_c(p)$ is zero if and only if $S_p$ is not minimal. Thus, the variety $V_{f_o f_c}$ is the variety we require.
$S_q$ is not controllable. Then, $S_q$ is not minimal, and so $q \in V_f$. Thus, $S$ is generically controllable. By the same argument, $S$ is generically observable.

To prove sufficiency, assume $S$ is generically controllable and generically observable. Then, there exist proper varieties $V_c$ and $V_o$ such that $p \in V_c$ if $S_p$ is not controllable and $p \in V_o$ if $S_p$ is not observable. Let $V_m$ be the variety \{ $p \in \mathbb{R}^n | c(p)o(p) = 0$ \}. Then, $p \in V_m$ if and only if $p \in V_c$ or $p \in V_o$. Thus, if $S_p$ is not minimal, either $p \in V_c$ or $p \in V_o$, and so $p \in V_m$. Since the set of all polynomial functions over the real numbers in $n$ indeterminates forms an integral domain, $c \cdot o \neq 0$, and so the variety $V_m$ is proper. Therefore, $S$ is generically minimal.

To prove Theorem 19, we first state graph-theoretical conditions for generic controllability. The following theorem states the conditions for generic controllability given by Dion et al.\cite{2}. Dion et al.\cite{2} note that similar results hold for generic observability. Unfortunately, we have not been able to find a suitable reference for these results.

**Theorem 19.** A structured linear system $S = (A, B, C, D)^S$ is generically controllable if and only if the following conditions hold:

1. Every state variable $x_i$ is the end vertex of some $U$-rooted simple path in $G(S)$.
2. There exists a disjoint union of a $U$-rooted simple path family and a cycle family that covers all state vertices.

To derive conditions for generic observability, we will first show that a structured linear system is generically observable if and only if its dual system is generically controllable. The dual system of a linear system and a structured linear system is stated below. To define the dual of a structured linear system, we also define the transpose of a structured matrix.

**Definition 34.** Let $S = (A, B, C, D)$ be a linear system. The dual system of $S$ is the system $D(S) = (A^T, C^T, B^T, D^T)$.

**Definition 35.** Let $A$ be a structured matrix determined by $T(A)$. The transpose $A^T$ of $A$ is the structured matrix determined by $T(A^T) = \{(j, i) | (i, j) \in T(A)\}$.

**Definition 36.** Let $S = (A, B, C, D)^S$ be a structured linear system. The dual system of $S$ is the structured linear system $D(S) = (A^T, C^T, B^T, D^T)^S$. Here, the transpose of a structured matrix defined in Definition 35 is used.

**Remark 6.** Let $S$ be a structured linear system. Then, for every system $S_q \in S$, there exists a system $D(S_q) \in \mathbb{D}^S(S)$ that is the dual system of $S_q$. However, due to the way we have defined the standard parameterization of a structured linear system, the vector $q$ is a permutation of the vector $p$.

The result that a structured linear system is generically observable if and only if its dual is generically controllable is stated below as a lemma. The proof of this result uses the well-known fact that a linear system is observable if and only if its dual is controllable and is deferred to an appendix.

**Lemma 18.** A structured linear system $S$ is generically observable if and only if its dual $D^S(S)$ is generically controllable.

To use the above result to derive graph-theoretical conditions for generic observability, we need to formally state the relation between the graph of a structured linear system and the graph of its dual. We state this relation in the lemma below. Intuitively, this lemma states that the graph of a dual system is obtained from that of the original system by interchanging the inputs and outputs of the system and reversing each arc of the original system’s graph. The proof of this lemma is straightforward but somewhat involved and is hence deferred to the appendix.

**Lemma 19.** Let $S$ be a structured linear system. Furthermore, let $G(D^S(S)) = (V_D, E_D)$ and $G(S) = (V_S, E_S)$. Then, the function $f : V_D \rightarrow V_S$ given below has the following properties:

1. $f$ is a bijection
2. For all $v_1, v_2 \in V_D$, $(v_1, v_2) \in E_D$ if and only if $(f(v_2), f(v_1)) \in E_S$

$$f(v) = \begin{cases} x_i(G(S)) & \text{if } v = x_i(G(D^S(S))) \\ y_i(G(S)) & \text{if } v = u_i(G(D^S(S))) \\ u_i(G(S)) & \text{if } v = y_i(G(D^S(S))) \end{cases}$$

In the lemmas below, we state the conditions that a system’s graph will satisfy if and only if the graph of the system’s dual satisfies the conditions for generic controllability. Using Lemma 18 it is clear that these conditions are a graph-theoretical characterization of generic observability. The proofs of these lemmas involve an intuitive application of Lemma 19. The details of these proofs are deferred to an appendix.

**Lemma 20.** Every state variable $x_i(G(S))$ is the first vertex of a $Y$-topped path if and only if every state variable $x_i(G(D^S(S)))$ is the end vertex of a $U$-rooted path.

**Lemma 21.** There exists a disjoint union of a $Y$-topped path family and a cycle family in $G(S)$ that covers every vertex $x_i$, if and only if a disjoint union of a $U$-rooted path family and a cycle family exists in $G(D^S(S))$ that covers every vertex $x_i$. 

(v. September 20, 2011, p.18)
Using the above Lemmas, we can state the graph-theoretical characterization of generic observability. The proof of this result is straightforward and is hence deferred to an appendix.

**Lemma 22.** $S$ is generically observable if and only if every vertex $x_i$ in $G(S)$ is the first vertex of a $Y$-topped path and there exists a disjoint union of a $Y$-topped path family and a cycle family in $G(S)$ that covers every vertex $x_i$.

Below, we complete the proof of Theorem 14.

**Proof of Theorem 7**. By Theorem 15, $S$ is generically minimal if and only if it is generically controllable and generically observable. Using Theorem 19, we find the first two conditions. The remaining conditions follow from Lemma 22.

Finally, we will show that the graph-theoretical conditions we have previously derived are a necessary condition for a given linear system to be minimal. We state the proof of this result below. Before we state this proof, we formally define the structured linear system corresponding to the graph $G(S)$ of a linear system $S$.

**Definition 37.** Let $S = (A, B, C, D)$ be a linear system. Furthermore, let $T(A) = \{(i, j) | A_{ij} = 0\}$ and similarly for $T(B), T(C)$ and $T(D)$. Then, let $A_i$ be determined by $T(A)$ and similarly for $B_s, C_s$ and $D_s$. The structured linear system corresponding to the graph of $S$ is the system $S_s = (A_s, B_s, C_s, D_s)^S$.

**Proof of Theorem 7**. Let $S$ be a minimal linear system and $S'$ the structured linear system corresponding to the graph $G(S)$. Since every entry of $S$ is zero if and only if the corresponding entry in $S'$ is a fixed zero, it is clear that the graphs $G(S)$ and $G(S')$ are identical. Furthermore, it is clear that $S \in S'$. Thus, since $S$ is minimal, $S'$ is generically minimal. Then, the graph $G(S')$ must satisfy the conditions of Theorem 14. Since the graphs $G(S)$ and $G(S')$ are identical, $G(S)$ must also satisfy these conditions, as claimed.

5 Conclusions

In our introduction, we described a number of studies in which researchers attempted to identify the structure of a dynamical system from input/output data. In this paper, we have considered to what degree this structure is determined by the input/output relation of a linear system.

We began by applying linear transformations to systems. As we saw in Subsection 3.3, such linear transformations may change the system’s graph structure. Furthermore, as we saw in Subsection 3.5, even the condensed graph of a system is not necessarily conserved by linear transformations. Finally, as we stated in Subsection 3.3, even the number of completely disconnected components in the system’s graph is not determined by input/output behavior.

The results described above indicate that many aspects of a system’s graph structure are not determined by its input/output relation. This implies that to identify the structure of a linear system from input/output data, we require assumptions about the system. We have considered two possible forms of such assumptions.

The first kind of assumption we considered was that the system under consideration has a particular canonical form. For instance, the system might be a minimal SISO system with a diagonal $A$-matrix. We showed in Subsection 3.3 that we can characterize the existence of (CG)-isomorphisms between two systems satisfying this kind of assumption. In addition, we showed in Subsection 3.5 that there exist certain conditions that must be satisfied by the graph of a minimal system.

The second kind of assumption we considered was that the system was a member of a particular structured system. That is, we assumed that some edges could not occur in the system’s graph. As we showed in Subsection 3.5, assumptions of this kind are not sufficient to uniquely identify a system’s parameters using input/output data.

To summarize, our results have two implications for the identification of a system’s structure. First, a system’s structure is not uniquely determined by the system’s input/output relation. Thus, we require additional assumptions to identify a system’s structure. Second, the assumption that a given set of edges does not occur in a system’s graph may be insufficient to identify a system’s structure. At the very least, this assumption is insufficient to uniquely identify the system parameters. Thus, even if the structure can be identified uniquely, the strength of the influence of one variable on another cannot be quantified.

5.1 Future work

As we remarked in the introduction, we conjecture that some of our results may apply, possibly in a modified form, to the models considered by Hollander’s and Friston et al. We also remarked that the relation between the graph structure of a vector AR model and the Granger causality criteria of Goebel et al. is still unclear. Thus, more work is needed to examine the implications of this paper for the models considered by Hollander, Friston et al. and Goebel et al.

A Appendix

This appendix consists of background material and technical proofs. In the first subsection, we briefly recall the
A.1 Elementary divisors and invariant polynomials

Consider an \( n \times n \) real matrix \( A \) and its characteristic matrix \( \lambda I - A \). Gantmacher [4, Ch. 6] shows that this characteristic matrix can be transformed to a diagonal matrix \( D \), as shown below, by elementary row and column operations. The diagonal elements of \( D \) are called the invariant polynomials of the characteristic matrix \( \lambda I - A \) or equivalently those of the matrix \( A \). An important property of the polynomials \( i_1, i_2, i_3, \ldots, i_n \) is that each divides the preceding one, that is \( i_1 = i_{i+1}p \), for some polynomial \( p \).

\[
D = \begin{bmatrix}
i_n & 0 & \cdots & 0 \\
0 & i_{n-1} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & i_1
\end{bmatrix}
\]

Gantmacher also shows an equivalent definition of these polynomials using minors of \( \lambda I - A \). Let \( D_i \) denote the greatest common divisor of the minors of order \( i \) of \( \lambda I - A \), with \( D_0 = 1 \). Then, we can equivalently define the invariant polynomials as ratios of these greatest common divisors, as follows:

\[
i_1 = \frac{D_n}{D_{n-1}}, i_2 = \frac{D_{n-1}}{D_{n-2}}, \ldots, i_n = \frac{D_1}{D_0}
\]

In the above definition of \( i_1 \), notice that \( D_n = |\lambda I - A| \). This implies that \( i_1 \) is the minimal polynomial of \( A \). As defined by Gantmacher [4, Ch. 4], this polynomial is the polynomial \( \psi \) of least degree such that \( \psi(A) = 0 \), where the first coefficient of \( \psi \) is taken to be 1.

Using the above definitions of the invariant polynomials, we can define the elementary divisors of \( A \). To do this, we factor each of the invariant polynomials \( i_j \) of \( A \) into powers of irreducible polynomials \( \phi_i \), as shown below. The powers of these polynomials \( \phi_i \) with exponents not equal to zero that appear in this factorization are called the elementary divisors of \( A \).

\[
i_1 = (\phi_1)^{k_{11}}(\phi_2)^{k_{12}}\cdots(\phi_m)^{k_{1m}}
\]
\[
i_2 = (\phi_1)^{k_{21}}(\phi_2)^{k_{22}}\cdots(\phi_m)^{k_{2m}}
\]
\[
\vdots
\]
\[
i_n = (\phi_1)^{k_{n1}}(\phi_2)^{k_{n2}}\cdots(\phi_m)^{k_{nm}}
\]

A.2 Technical proofs: Systems, graph structures and equivalent structures

Proof of Observation 2 For input vertices, it is clear that no path from a vertex \( v \) to the input \( u_i(G(S)) \) exists, as input vertices have indegree zero. Thus, there exist no vertices \( v \) other than \( u_i(G(S)) \) such that \( v \leftrightarrow u_i(G(S)) \). Therefore, \( u_i(G(S)) \) is the sole member of its component. The proof for \( y_i(G(S)) \) is similar.

A.3 Technical proofs: Graph isomorphism and its inadequacy

Proof of Lemma 2 To prove the first condition, consider the identity function on \( V(G(S)) \). Clearly, the identity is bijective and satisfies the second condition of Definition 7. It is also clear that the identity is type-restricted, and so \( S \cong S \).

To prove symmetry, let \( S \cong S' \). Then, there exists a type-restricted isomorphism \( \phi : V(G(S)) \rightarrow V(G(S')) \). Clearly, the inverse \( \phi^{-1} \) is bijective. This inverse also satisfies the second condition of Definition 7 since \( \phi \) is surjective. For the same reason, the type-restriction condition must also be satisfied. Therefore, \( S' \cong S \).

To complete the proof, let \( S \cong S' \) and \( S'' \cong S'' \). Then, there exist type-restricted isomorphisms \( \phi_1 : V(G(S)) \rightarrow V(G(S')) \) and \( \phi_2 : V(G(S')) \rightarrow V(G(S'')) \). The composition \( \phi = \phi_2 \circ \phi_1 \) is then a bijection from \( V(G(S)) \) to \( V(G(S'')) \). This composition also satisfies the other conditions for a type-restricted isomorphism, as can be readily verified.

Proof of Lemma 3 Since the rows of \( P(e_1, e_2, \ldots, e_n) \) are a permutation of the rows of \( I_n \), they are clearly an orthogonal set of unit vectors. Therefore, \( P(e_1, e_2, \ldots, e_n) \) is clearly orthogonal.

A.4 Technical proofs: Condensed-graph isomorphism

Proof of Lemma 5 Let \( v_1, v_2, \ldots, v_m \) be the indices of the state variables in the trap in the graph of \( G(S) \). Then, it is clear that \( (x_{v_i}, y_j) \) is not an edge of \( G(S) \) for all \( i \) and \( j \), or equivalently, that \( C_{jv_i} = 0 \). For the same reason, for all integers \( j \) such that \( j \neq v_i \) for all \( i \), \( A_{jv_i} = 0 \) for all \( i \). Consider \( (CA)_{jv_i} = \sum_k C_{jk}A_{kv_i} \). From the above discussion, we have for all \( k \) that either \( C_{jk} = 0 \) or \( A_{kv_i} = 0 \). Thus, \( (CA)_{jv_i} = 0 \). By repeating this argument, we can show that the same holds for \( CA^k \) for all \( k \). Therefore, the columns \( v_1, v_2, \ldots, v_m \) of \( D \) are zero. Since \( D \) has \( n \) columns, \( D \) cannot be of rank \( n \).

Proof of Lemma 7 Let \( i \neq |\lambda I - A| \). Since \( |\lambda I - A| = i \gcd(\text{adj}(\lambda I - A)) \), where \( \text{adj}(\lambda I - A) \) is the adjugate of \( \lambda I - A \), \( \gcd(\text{adj}(\lambda I - A)) \neq 1 \). Consider the nominal transfer function \( H = \frac{C_{\text{adj}(\lambda I - A)B}}{|\lambda I - A|} \). Since
\[ |M - A| = \gcd(\text{adj}(M - A))i_1 \text{ and } \text{adj}(M - A) = \frac{\gcd(\text{adj}(M - A))\Gamma}{i_1} \text{ for some matrix } \Gamma, \quad H = \frac{CTB}{i_1}. \] Thus, \( H \) has a pole-zero cancellation and so the realization \((A, B, C, D)\) is non-minimal.

**Proof of Lemma 8.** Since \( \phi \) is irreducible, either \( \phi(0) \neq 0 \) or \( \phi(\lambda) = \lambda^n \) for some \( m \geq 1 \). In the former case, this implies that in \( \phi(\lambda) = \lambda^n + a_1\lambda^{n-1} + \cdots + a_m, a_m \neq 0 \). Therefore, the top-right entry in the \( n \times n \) companion matrix \( L \) corresponding to \( \phi(\lambda)^k \) is non-zero. Then, an edge \((x_i, x_{i+1})\) exists in the graph of \( L \). Since the fixed elements equal to 1 on \( L \)'s subdiagonal correspond to edges \((x_i, x_{i+1})\) for \( 1 \leq i < n \), this graph has a Hamiltonian cycle.

Otherwise, if \( \phi(\lambda) = \lambda^n \), then the last column is zero. Then, the only edges that exist imply that this graph is a directed path.

**Proof of Lemma 11.** Since the realization \( S \) is minimal, every vertex in every component cannot be part of either an unreachable set or a trap. In the first case, if the component is Hamiltonian, if no edge of the form \((u, x_i)\) exists for \( x_i \) in the component, the entire component is unreachable. Similarly, if no edge of the form \((x_j, y)\) exists for \( x_j \) in the component, the entire component is a trap. Therefore, edges of these forms must exist. Since the component is Hamiltonian, these conditions are also sufficient for the component to contain neither traps nor unreachable sets.

In the second case, the component is a directed path consisting of the vertices \( x_i, x_{i+1}, \cdots, x_{i+m} \). If the edge \((u, x_i)\) does not exist in \( G(S) \), \( \{x_i\} \) is an unreachable set, and so \( S \) is non-minimal. Similarly, if \((x_{i+m}, y)\) does not exist in \( G(S) \), \( \{x_{i+m}\} \) is a trap, and so \( S \) is non-minimal. Thus, the stated conditions must hold. Furthermore, if the conditions are satisfied, paths from \( u \) to \( x_i \) and from \( x_i \) to \( y \) exist in \( G(S) \) for all \( x_i \) in the component, and so the component does not contain traps or unreachable sets.

By Lemma 8, it is clear that each component consisting of state variables in \( G(S) \) is covered by one of the two cases above. Therefore, no such component can contain traps or unreachable sets, and so \( G(S) \) can contain neither traps nor unreachable sets.

**Proof of Lemma 10.** Let \( A \) have two elementary divisors \( \phi(\lambda)^{k_1} \) and \( \phi(\lambda)^{k_2} \). Since these elementary divisors are powers of the same irreducible polynomial, they cannot occur in the same invariant polynomial. After all, if they did, this invariant polynomial would only have a single elementary divisor \( \phi(\lambda)^{k_1+k_2} \). Therefore, two or more invariant polynomials of \( A \) are not equal to 1. Then, the minimal polynomial \( i_1 \) of \( A \) cannot coincide with \( |M - A| \), and so by Lemma 7 \( S \) is non-minimal.

**A.5 Technical proofs: Components of condensed graphs**

**Lemma 23.** Let \( \phi_i, 1 \leq i \leq l \) be the irreducible polynomials that divide \( |M - A| \). Furthermore, let \( E_i, 1 \leq i \leq m \) be sets of elementary divisors of \( A \) such that no two elements of \( E_i \) are of the form \( \phi_j^l \) for the same \( j \), for all \( i \). Additionally, let each elementary divisor of \( A \) be an element of exactly one set \( E_i \). Then, the \( m \)-tuple \((E_1, E_2, \cdots, E_m)\) corresponds to a block-companion realization similar to \( S = (A, B, C, D) \).

**Proof.** Let \( l_i = \prod_{c \in E_i} c, 1 \leq i \leq m \), and \( L_i \) be the companion matrix corresponding to \( l_i \). We claim that \( A' = \{L_i\} \) is similar to \( A \). Since \( A' \) is block-diagonal, the elementary divisors of \( A' \) are those of the matrices \( L_i \), by Theorem 13. Since each of the matrices \( L_i \) is a companion matrix corresponding to the product of the elements of \( E_i \), the elementary divisors of \( L_i \) are the elements of \( E_i \). Thus, each elementary divisor of \( A \) is an elementary divisor of \( A' \), since we require that each such divisor is an element of some \( E_i \). In addition, \( A' \) can have no other elementary divisors, as each element of each set \( E_i \) is an elementary divisor of \( A \) and each elementary divisor of \( A \) is in exactly one set \( E_i \). Therefore, \( A' \) and \( A \) are similar. Thus, there exists a matrix \( T \) such that \( A' = TAT^{-1} \). Therefore, \( T(S, T) \) is a block-companion realization similar to \( S \), as claimed.

**Proof of Lemma 11.** Let \( \phi_i, 1 \leq i \leq m \) be the irreducible polynomials that divide \( |M - A| \). Furthermore, let \( e_{ij} \) be the \( j \)-th elementary divisor of \( A \) of the form \( \phi_i^l \), in some arbitrary order. Let \( E_{ij} \) be a set of elementary divisors of \( A \), defined as follows:

\[ E_{ij} = \begin{cases} \{e_{ij}\} & \text{if } e_{ij} \text{ exists} \\ \emptyset & \text{otherwise} \end{cases} \]

Clearly, each elementary divisor \( e_{ij} \) is an element of only \( E_{ij} \) and no other set \( E_{in} \). Thus, the \( k \)-tuple \( t = (\cup_{1 \leq i \leq m} E_{ij}, \cup_{1 \leq i \leq m} E_{i2}, \cdots, \cup_{1 \leq i \leq m} E_{ik}) \) consists of sets of elementary divisors of \( A \). It is clear that each elementary divisor of \( A \) is an element of exactly one element of the tuple \( t \). Furthermore, each element of the tuple \( t \) contains at most 1 elementary divisor of the form \( \phi_i^l \) for each polynomial \( \phi_i \). Thus, by Lemma 23, \( t \) corresponds to a realization \( S' \) similar to \( S \). \( S' \) is a block-companion realization with \( k \) diagonal blocks, as claimed.

**Proof of Lemma 12.** Let \( A' \) be the second natural normal form of \( A \). Then, \( A' \) has \( n \) diagonal blocks and is a block-companion matrix. Furthermore, \( A' \) is similar to \( A \). Thus, a matrix \( T \) exists such that \( A' = TAT^{-1} \). Therefore, \( T(S, T) \) is the required realization.
Proof of Lemma 14. Let \((E_i)\) be an \(l\)-tuple of sets of elementary divisors, where \(E_i\) consists of the elementary divisors of the \(i\)-th diagonal block of \(A\). Assume \(l \neq n\). Then, since each elementary divisor of \(A\) is a member of some set \(E_i\), at least one set \(E_i\) consists of two or more elements. Let \(j\) be an integer such that \(E_j\) consists of two or more elements and select an arbitrary element \(e\) of \(E_j\). We claim that \(t' = (E_1, \ldots, E_{j-1}, E_j \setminus \{e\}, E_{j+1}, \ldots, E_l, \{e\})\) is an \((l+1)\)-tuple satisfying the conditions of Lemma 23. Clearly, since none of the sets \(E_i\) contains two elementary divisors of the form \(\phi_i\) for some irreducible polynomial \(\phi_i\), neither do the elements of \(t'\). The elements of \(t'\) are also clearly sets of elementary divisors of \(A\). Furthermore, since each elementary divisor is an element of exactly one set \(E_i\), the same holds for the elements of \(t'\). Thus, by Lemma 23 the tuple \(t'\) corresponds to a block-companion realization similar to \(S\) with \(l+1\) diagonal blocks.

A.6 Technical proofs: Structured systems and their graphs

Proof of Lemma 14. Note that \(T(S,M) = (MAM^{-1}, MB, CMA^{-1}, D)\). First, we show by induction that \(x_{T(S,M),k} = Mx_{S,k}\). Since we use zero initial conditions, \(x_{T(S,M),k} = 0 = M0 = Mx_{S,0}\). Inductively, \(x_{T(S,M),k+1} = MAM^{-1}x_{T(S,M),k} + MBu_k = MAM^{-1}Mx_{S,k} + Bu_k = M(Ax_{S,k} + Bu_k) = Mx_{S,k+1}\). Therefore, \(y_{T(S,M),k} = CM^{-1}x_{T(S,M),k} + Du_k = Cx_{S,k} + Du_k = y_{S,k}\).

Proof of Lemma 10. Since \(P\) is generic for \(S\), there exists a proper variety \(V_p\) such that \(\{p \in \mathbb{R}^n | P \text{ does not hold for } S_p\} \subset V_p\). Suppose \(\neg P\) is generic for \(S\), i.e. there exists a proper variety \(V_g\) such that \(\{p \in \mathbb{R}^n | P \text{ holds for } S_p\} \subset V_g\). Then, \(\mathbb{R}^n \subset V_f \cup V_g\) and so \(V_f \cup V_g = \mathbb{R}^n\). Let \(h(x) = f(x)g(x)\). Then, \(V_f \cup V_g = V_h = \mathbb{R}^n\). Therefore, \(h\) is the zero function. But then, since the domain of polynomials over \(\mathbb{R}\) in \(n\) indeterminates is an integral domain, either \(f\) or \(g\) must be the zero function. Since \(V_f\) is proper, \(f\) is non-zero for some \(x\), and so \(g\) must be the zero function. But then, either \(V_g\) is not proper, which is impossible.

Lemma 24. Let \(S = (A, B, C, D)\) be a linear system. Then, \(S\) is observable if and only if \(\mathbb{D}(S)\) is controllable.

Proof. This result is implicitly given in the textbook by Kailath [7]. To formally prove it, notice that \(\mathbb{D}(S) = (A^T, Ct, B^T, D^T)\). Thus, the controllability matrix of \(\mathbb{D}(S)\) is \(C_{D} = \begin{bmatrix} C \\ CA \\ \vdots \\ C A^{n-1} \end{bmatrix}\). We notice that \(C_D = G_S^T\). Thus, these matrices have the same rank, completing our proof.

Proof of Lemma 15. To prove necessity, assume \(S\) is generically observable. Thus, there exists a proper variety \(V_f\) such that if \(S_p\) is not observable, \(p \in V_f\). Let \(g\) be the polynomial obtained from \(f\) by permuting the indeterminates in \(f\) such that for all \(p\) and \(q\) such that \(D(S_p) = \mathbb{D}(S_q)\), \(f(p) = 0\) if and only if \(g(q) = 0\). Let \(g\) be an arbitrary parameter vector such that \(\mathbb{D}(S_p) = \mathbb{D}(S_q)\). Then, \(S_p\) is not observable, and so \(p \in V_f\). But then, \(q \in V_g\). Since \(f\) is not identically zero, neither is \(g\), and so \(V_f\) is proper. Thus, \(\mathbb{D}(S)\) is generically controllable.

The sufficiency of the condition follows from a similar argument.

Proof of Lemma 19. The first property, that \(f\) is a bijection, is clear from the definition of \(f\) and the definition of the dual \(\mathbb{D}(S)\).

To prove the second property, let \(S = (A, B, C, D)\). Then, \(\mathbb{D}(S) = (AT, CT, BT, DT)^S\). We will consider all the types of edges that occur in \(\mathbb{D}(S)\). First, consider edges of the form \((x_i, x_j)\), which exist in \(\mathbb{D}(S)\) if and only if \(AT_{ij}\) is not a fixed zero. Since \(AT_{ij}\) is a fixed zero if and only if \(A_{ij}\) is a fixed zero, this edge exists if and only if \((x_i, x_j)\) is an edge of \(G(S)\). Since \(f(x_i) = x_i\), this proves our condition for edges of this form.

Second, consider edges of the form \((u_i, x_j)\), which exist in \(\mathbb{D}(S)\) if and only if \(C_{ij}\) is not a fixed zero. Since \(C_{ij}\) is a fixed zero if and only if \((x_i, y_j)\) is an edge of \(G(S)\). Since \(f(u_i) = y_i\) and \(f(x_i) = x_j\), this proves our condition for edges of this type.

The arguments for the remaining edges are similar.

Proof of Lemma 20. To prove necessity, let \(x_1 v_1 v_2 \cdots v_k y_j\) be a \(Y\)-tipped path in \(G(S)\). Let \(f\) be the mapping of Lemma 19. Then, since \(f\) is a bijection and the vertices \(x_1, v_i\) and \(y_j\) are all distinct, so are the vertices \(f^{-1}(x_1), f^{-1}(v_i)\) and \(f^{-1}(y_j)\). Furthermore, since edges \((x_1, v_i), (v_i, v_{i+1})\) and \((v_k, y_j)\) exist in \(G(S)\), edges \((f^{-1}(v_1), f^{-1}(v_i)), (f^{-1}(v_{i+1}), f^{-1}(v_k))\) and \((f^{-1}(y_j), f^{-1}(v_k))\) exist in \(\mathbb{D}(S)\). Thus, using the definition of \(f\), we find that \(u_i v_i v_2 \cdots v_k x_i\) is a path in \(\mathbb{D}(S)\), where \(v_i = f^{-1}(v_i)\).

The sufficiency of the condition follows from the same argument.
To prove necessity, suppose a disjoint union as described above exists. Then, every state variable $x_i$ is covered by either a $Y$-topped path or a cycle. Using the proof of Lemma 20, a state variable covered by a $Y$-topped path is covered by a $U$-rooted path in $G(DS(S))$. A similar argument shows that a state variable covered by a cycle in $G(DS(S))$ will also be covered by a cycle in $G(DS(S))$. Thus, every $Y$-topped path in the path family corresponds to a $U$-rooted path in $G(DS(S))$, and every cycle corresponds to a cycle. It remains to be shown that these paths and cycles are mutually disjoint. Thus, let $v_1v_2 \cdots v_k$ and $v'_1v'_2 \cdots v'_k$ be disjoint paths or cycles in $G(S)$. Then, the corresponding paths or cycles in $G(DS(S))$ are given by $f^{-1}(v_k) \cdots f^{-1}(v_2)f^{-1}(v_1)$ and $f^{-1}(v'_k) \cdots f^{-1}(v'_2)f^{-1}(v'_1)$. Suppose these paths are not disjoint. Then, there exist $i$ and $j$ such that $f^{-1}(v_i) = f^{-1}(v'_j)$. Since $f^{-1}$ is injective, this implies that $v_i = v'_j$. This contradicts our assumption that the paths $v_1v_2 \cdots v_k$ and $v'_1v'_2 \cdots v'_k$ were disjoint. Thus, the corresponding paths in $G(DS(S))$ must be disjoint.

A similar argument in the other direction proves the sufficiency of the condition.

Proof of Lemma 22. By Lemma 18, $S$ is generically observable if and only if the dual $DS(S)$ is generically controllable. Furthermore, by Theorem 19, $DS(S)$ is generically controllable if and only if every vertex $x_i$ in $G(DS(S))$ is the end vertex of a $U$-rooted path and there exists a disjoint union of a $U$-rooted path family and a cycle family in $G(DS(S))$ that covers every vertex $x_i$. By Lemmas 20 and 21, these conditions are equivalent to the stated conditions on $G(S)$, completing the proof.

References

[1] Chartrand, Gary and Zhang, Ping (2005). Introduction to Graph Theory. McGraw-Hill.
[2] Dion, Jean-Michel, Commault, Christian, and Woude, Jacob van der (2003). Generic properties and control of linear structured systems: a survey. Automatica, Vol. 39, pp. 1125–1144.
[3] Friston, K.J., Harrison, L., and Penny, W. (2003). Dynamic causal modeling. Neuroimage, Vol. 19, pp. 1273–1302.
[4] Gantmacher, F.R. (1959). Theory of Matrices, Vol. 1. Chelsea Publishing Company, New York, N.Y.
[5] Goebel, Rainer, Roebroeck, Alard, Kim, Dae-Shik, and Formisano, Elia (2003). Investigating directed cortical interactions in time-resolved fmri data using vector autoregressive modeling and Granger causality mapping. Magnetic Resonance Imaging, Vol. 21, pp. 1251–1261.
[6] Hollanders, Goele (2009). Identifying Dynamic Sparse Interaction Networks. Ph.D. thesis, Universiteit Maastricht and Universiteit Hasselt.
[7] Kailath, Thomas (1980). Linear Systems. Prentice-Hall Inc.,Eaglewood Cliffs, N.J.
[8] Vaccaro, Richard J. Digital Control A State-Space Approach. McGraw-Hill,Inc.