2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Connecting graphical notions of separation and statistical notions of independence for topology reconstruction

Donatello Materassi (mater013@umn.edu)
Department of Electrical Engineering and Computer Engineering, University of Minnesota

Abstract— **Over the last decade, there has been a significant increase in interest for techniques that can infer the connectivity structure of a network of dynamic systems. This article examines a flexible class of network systems and reviews various methods for reconstructing their underlying graph. However, these techniques typically only guarantee consistent reconstruction if additional assumptions on the model are made, such as the network topology being a tree, the dynamics being strictly causal, or the absence of directed loops in the network. The central theme of the article is to reinterpret these methodologies under a unified framework where a graphical notion of separation between nodes of the underlying graph corresponds to a probabilistic notion of separation among associated stochastic processes.**

I. Introduction

Over the past decade, there has been a growing interest in methodologies that can learn the unknown structure of a network of dynamic systems from observed time series [1], [2], [3], [4], [5], [6], [7]. These techniques cover a range of scenarios, including the use of appropriately designed inputs to recover the network topology, the knocking-out of nodes, and assuming that the network is forced by unknown, non-manipulable inputs. This article focuses on the last scenario, known as network reconstruction under non-invasive or passive observations. In this case, it is assumed that the network is operating under standard conditions, being excited by unknown forcing inputs, and only the outputs corresponding to the network nodes are observable. The article revisits previously known results for the non-invasive identification of Linear Dynamic Influence Models (LDIMs) and reinterprets these techniques under a unifying framework.

Specifically, it shows that the techniques rely on a duality between a probabilistic notion of separation defined on the observed time series and a graphical notion of separation based on the graph representing the network connectivity. Once this duality is established, appropriate algorithms can be created to obtain a consistent reconstruction. The article also attempts to draw connections with other models, such as Dynamic Structure Functions [8], dynamic Bayesian networks [9] and other input output description [10], [11], that are semantically close or equivalent to LDIMs.

Notation

We use the following notation:

- $e_1, ..., e_n$: unobserved wide-sense stationary signals;
- $y_1, ..., y_n$: observed wide-sense stationary signals;
- $H(\mathbf{z})$: transfer matrix
- Φ_x: Power Spectral Density of a signal x;
- Φ_{xw}: Cross-Power Spectral Density of x and w.

Also, we use the following notation for subvectors. *Index subset notation for vectors*
Let $v^T := (v_1^T|...|v_n^T)$ be a vector defined by n subvectors, $v_1, ..., v_n$, and let $I := (i_1, ..., i_{n_I})$ be an ordered set of integers in $\{1, ..., n\}$. We denote by $v_I^T := (v_{i_1}^T|...|v_{i_{n_I}}^T)$, the vector obtained by considering the subvectors in v indexed by $(i_1, ..., i_{n_I})$.

II. Preliminary notions and problem formulation

In this section we introduce some preliminary notions for the definition of a class of networks referred to as Linear Dynamic Influence Models (LDIMs).

A. Basic notions of graph theory

Directed and undirected graphs are defined as follows.

Definition 1 (Directed and Undirected Graphs): A *directed graph* G is a pair $(V, \vec{E})$ where V is a set of vertices (or nodes) and $\vec{E}$ is a set of edges (or arcs) which are ordered pairs of elements of V. An *undirected graph* G is a pair $(V, \bar{E})$ where V is a set of vertices (or nodes) and $\bar{E}$ is a set of edges (or arcs) which are unordered pairs of elements of V. □

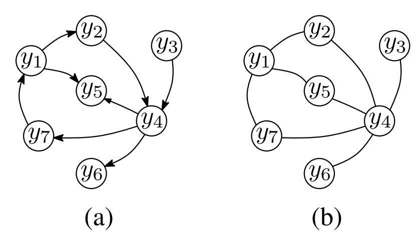

Fig. 1. A directed graph. (a) and its skeleton (b).

On directed graphs we also define "chains" and "paths".

Definition 2 (Paths, Chains): Consider a directed graph $G = (V, \vec{E})$ with vertices $y_1, ..., y_n$. A *chain* starting from y_i and ending in y_j is an ordered sequence of distinct edges in $\vec{E}$ given by $((y_{\pi_1}, y_{\pi_2}), (y_{\pi_2}, y_{\pi_3}), \ldots, (y_{\pi_{\ell-1}}, y_{\pi_\ell}))$ where $y_i = y_{\pi_1}$, $y_j = y_{\pi_\ell}$, and $(y_{\pi_p}, y_{\pi_{p+1}}) \in \vec{E}$ for all $p = 1, ..., \ell - 1$. A *path* between two vertices y_i and y_j is an ordered sequence of distinct ordered pairs of nodes

$$((y_{\pi_1}, y_{\pi_2}), (y_{\pi_2}, y_{\pi_3}), \ldots, (y_{\pi_{\ell-1}}, y_{\pi_\ell}))$$

where $y_i = y_{\pi_1}$, $y_j = y_{\pi_\ell}$, and either $(y_{\pi_p}, y_{\pi_{p+1}}) \in \vec{E}$ or $(y_{\pi_{p+1}}, y_{\pi_p}) \in \vec{E}$ for all $p = 1, ..., \ell - 1$. □

As it follows from the definition, chains are a special case of paths. All paths (and consequently also all chains) can be suggestively denoted by separating the nodes in the sequence $\{y_{\pi_p}\}_{p=1}^{\ell}$ with the arrow symbol $\rightarrow$ if $(y_{\pi_{p-1}}, y_{\pi_p}) \in \vec{E}$ or the symbol $\leftarrow$ if $(y_{\pi_p}, y_{\pi_{p-1}}) \in \vec{E}$. For example, in Fig. 1(a), the path $y_1 \rightarrow y_2 \rightarrow y_4 \rightarrow y_6$ is also a chain, while $y_1 \rightarrow y_5 \leftarrow y_4 \leftarrow y_3$ is a path, but not a chain. Here we also say that, there is a chain from vertex y_1 to vertex y_6 in the graph.

From the concept of chain, we can derive the notions of ancestry and descendance.

Definition 3 (Parents, children, ancestors, descendants): Consider a directed graph $G = (V, \vec{E})$. A vertex y_i is a *parent* of a vertex y_j if there is a directed edge from y_i to y_j. In such a case y_j is a *child* of y_i. Also y_i is an *ancestor* of y_j if $y_i = y_j$ or if there is a chain from y_i to y_j. In such a case y_j is a *descendant* of y_i. Given a set $y_I \subseteq V$, we define the following sets:

$$pa_G(y_I) := \{w \in V \mid \exists y \in y_I : w \text{ is a parent of } y\},$$
$$ch_G(y_I) := \{w \in V \mid \exists y \in y_I : w \text{ is a child of } y\},$$
$$an_G(y_I) := \{w \in V \mid \exists y \in y_I : w \text{ is an ancestor of } y\}, \text{ and}$$
$$de_G(y_I) := \{w \in V \mid \exists y \in y_I : w \text{ is a descendant of } y\}.$$

We also define forks and colliders that ware certain substructures of three consecutive nodes in a path.

Definition 4 (Forks and colliders): A path involving the nodes $y_{\pi_1}, ..., y_{\pi_\ell}$ has a *fork* at y_{π_p}, for $1 < p < \ell$, if $y_{\pi_{p-1}}$ and $y_{\pi_{p+1}}$ are both children of y_{π_p} (that is $y_{\pi_{p-1}} \leftarrow y_{\pi_p} \rightarrow y_{\pi_{p+1}}$ appears in the path). A path has an *inverted fork* (or a collider) at y_{π_p} if $y_{\pi_{p-1}}$ and $y_{\pi_{p+1}}$ are both parents of y_{π_p} (that is $y_{\pi_{p-1}} \rightarrow y_{\pi_p} \leftarrow y_{\pi_{p+1}}$ appears in the path).

B. Wiener Filtering

The theory of Wiener filtering [12] provides a convenient way to represent, in the limit of infinite data, the outcome of linear regressions for the least square estimation of a stochastic process using a set of other stochastic processes. In this article we limit ourselves to causal Wiener filtering.

Definition 5 (Stable & Causal Transfer Functions): We define $\mathcal{F}^+$ as the space of causal transfer functions that are real-rational with domain of convergence that includes the complex unit circle. Namely, $H(\mathbf{z}) \in \mathcal{F}^+$ if there exists a sequence $\{h_k\}_{k=0}^{+\infty}$

$$H(\mathbf{z}) = \sum_{k=0}^{+\infty} h_k \mathbf{z}^{-k} \qquad \text{for all } |\mathbf{z}| = 1.$$

Transfer functions can operate on stochastic processes to definine linear spaces.

Definition 6 (Causal Transfer Function Space (ctfspan)): Consider a vector of jointly wide-sense stationary signals $y = (y_1, ..., y_n)$ with rational power spectral density Φ_y. The causal transfer function span (ctfspan) is defined as

$$\text{ctfspan}\{y_1, ..., y_n\} := \left\{ q = \sum_{i=1}^{n} Q_i(\mathbf{z}) y_i \mid Q_i(\mathbf{z}) \in \mathcal{F}^+ \right\}.$$

We provide a specific formulation of the causal Wiener filter.

Proposition 2.1: (*Causal Wiener Filter for processes with rational Power Spectral Density*) Let x and $y_1, ..., y_n$ be vector processes that are jointly stationary with rational power cross-spectral densities and rational auto-spectral densities. Define $y := (y_1^T | ... | y_n^T)^T$ and

$$X := \text{ctfspan}\{y_1, ..., y_n\}.$$

Consider the optimization problem

$$\inf_{q \in X} \mathbf{E}[(x - q)^T (x - q)] \tag{1}$$

where, $\mathbf{E}[\cdot]$, is the expectation operator. If $\Phi_y(e^{i\omega}) > 0$, for $\omega \in [-\pi, \pi]$, then the solution $\hat{x}_y$ for (1) exists and is unique and there is a unique rational transfer function $W_{x|y}(z)$, referred to as causal Wiener Filter, such that

$$\hat{x}_y = W_{x|y}(z) y.$$

Moreover, $\hat{x}_y$ is the only element in X such that, for any $q \in X$, $\mathbf{E}[(x - \hat{x}_y) q] = 0$.

Proof: This is a result from standard Wiener filtering theory. See [13]. ■

Wiener filter $W_{x|y}$ estimating the scalar signal x from y is a row vector of transfer functions. In order to indicate the component of $W_{x|y}$ operating on some elements y_I of y, we introduce the following notation.

Definition 7: (Component notation for Wiener filter) Let $y = (y_1, ..., y_n)$ and $I \subseteq \{1, ..., n\}$. We denote with y_I the subvector of y with elements indexed by I. We use the notation $W_{x,[y_I]|y}$ to denote the vector of components of the Wiener filter associated with the vector of signals y_I in the estimation of x. According to this notation, we have

$$W_{x|y} = \left(W_{x,[y_1]|y}, W_{x,[y_2]|y}, \dots, W_{x,[y_n]|y} \right)$$

C. Linear Dynamic Influence Models

We now introduce a specific class of dynamic networks defined by input-output relations among wide-sense stationary stochastic processes.

Definition 8 (Linear Dynamic Influence Models): A Linear Dynamic Influence Model (LDIM) is defined as a pair $(H(\mathbf{z}), e)$ where

- $e = (e_1^T | ... | e_n^T)^T$ is a vector of n wide-sense stationary stochastic vector processes such that the cross-spectral density matrix, $\Phi_{e_i e_j} = 0$, for $i \neq j$, and the auto-spectral density matrix, $\Phi_{e_j}(\mathbf{z})$, is rational and has full rank for all $|\mathbf{z}| = 1$ for $j = 1, ..., n$. Thus, the power spectral density, Φ_e, is block diagonal and has full rank for all $|\mathbf{z}| = 1$.
- $H(\mathbf{z})$ is a $n \times n$-block matrix of rational and stable transfer functions with $H_{ji}(\mathbf{z})$ being the (j, i) block-entry of $H(\mathbf{z})$ where stability is given by the absence of poles on the unit-circle.

The output processes, y_j, for $j = 1, .., n$ of the LDIM are defined as

$$y_j = e_j + \sum_{i=1}^{n} H_{ji}(\mathbf{z}) y_i, \tag{2}$$

or in a more compact way $y = e + H(\mathbf{z}) y$, where $y = (y_1^T | ... | y_n^T)^T$. We say that the LDIM is well-posed if $(\mathcal{I} - H_{II}(\mathbf{z}))$ is invertible for all $I \subseteq \{1, ..., n\}$. The well-posedness of a LDIM guarantees that y is wide-sense stationary.

In a more informal way, a LDIM is a network of stochastic vector processes $y_1, ..., y_n$ interconnected with each other via input-output relations defined by the transfer functions populating the transfer matrix $H(\mathbf{z})$, where the vector processes are also excited by independent stochastic process.

Definition 9: (*Graphical representations of a LDIM and its associated perfect directed graph (PDG)*) Given a LDIM, $(H(\mathbf{z}), e)$, with output processes $y = (y_1^T | ... | y_n^T)^T$ and a directed graph $G = (V, \vec{E})$ where $V = \{y_1, ..., y_n\}$ we say that G is a graphical representation of G if $(y_i, y_j) \notin \vec{E}$ implies $H_{ji}(\mathbf{z}) = 0$. If it also holds that $H_{ji}(\mathbf{z}) = 0$ implies $(y_i, y_j) \notin \vec{E}$, then we say that G is the (unique) Perfect Directed Graph (PDG) associated with the LDIM.

Abusing the nomenclature we will sometimes refer to nodes, edges, paths and chains of a LDIM even though, formally, we should refer to them as nodes, edges, paths and chains of its graphical representation or its PDG.

D. Connections with other models in the literature

The representation of the structure of a set of mathematical relations via graphs has been used in several areas of science. In particular it has been pioneered by Sewall Wright to represent Structural Equation Models (SEMs) in statistics [14], [15], [16] and since then has found widespread use in the social sciences [17], [18], [19]. More recently SEMs have been reinterpreted in terms of factorization of probability distributions and generalized in the form of "graphical models" [20], [21]. However, SEMs and graphical models only involve scalar static variables, so no time variable is considered and there is no notion of dynamics. As a consequence loops are not admitted in the resulting graph representations.

LDIMs and their graphical representations (introduced under the name of Linear Dynamic Graphs [22]) can be seen as a linear extension of SEMs to include dynamics. Thus, LDIMs are semantically equivalent to gaussian dynamic Bayesian networks [9]. At the same LDIMs are very similar to Signal Flow Graphs (SFGs) popularized by Mason [11], [23] even though SFGs specifically deal with deterministic signals and do not consider implicitely noise on each node. The Dynamic Function Structure representation [8] defined on the state-space to visualize the dynamic relation between the subset of observed components of the state is another very similar class of models with the only fundamental difference that considers only strictly causal transfer functions. The network model introduced in [10] is again semantically equivalent to LDIMs and gaussian dynamic Bayesian networks as well, but explicitly takes into account both manipulable inputs and noise components. The results in [10] are mostly about the identification of individual transfer function and already assume the knowledge of the network structure.

E. Problem Formulation

Assume that the output $y = (y_1, ..., y_n)$ of a LDIM is observed (equivalently the PSD Φ_y is known or estimated). Find the graphical representation of the LDIM.

III. LDIMs with polytrees structure

Polytrees, often referred to as "directed trees", are a class of directed graphs which can be formally described as follows.

Definition 10 (Polytrees): A polytree is a directed graph such that there exists a unique path connecting each pair of its nodes.
Figure 2 shows an instance of a polytree.

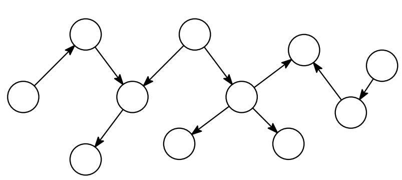

Fig. 2. An example of polytree.

The focus of this section is on techniques allowing for the reconstruction of Linear Dynamic Polytrees.

Definition 11 (Linear Dynamic Polytree): We say that a LDIM is a Linear Dynamic Polytree (LDP) if its graphical representation is a polytree.

A technique for the consistent reconstruction of a LDP from observed time series data obtained from its nodes is described in [24]. The main result is based on the definition of the following distance between stochastic processes

Definition 12 (Coherence Metric): Given two jointly wide-sense stationary processes y_i and y_j we define the distance

$$d(y_i, y_j) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left(1 - \frac{\left|\Phi_{y_i y_j}(e^{i\omega})\right|^2}{\Phi_{y_i}(e^{i\omega})\Phi_{y_j}(e^{i\omega})}\right)d\omega. \quad (3)$$

The introduction of a distance on a set of stochastic processes makes such a set a metric space where a notion of separation can be defined as follows.

Definition 13 (Neighbor-separation): Given a metric space with a distance $d(\cdot,\cdot)$, two elements y_i and y_j in the metric space are neighbor-separated by a point y_k if

$$d(y_i, y_k) < d(y_i, y_j) \text{ and}$$
$$d(y_j, y_k) < d(y_i, y_j).$$

We denote this relation as $nsep(y_i|y_k|y_j)$.
In other words, two elements y_i and y_j are separated by y_k if the distance of y_k from both y_i and y_i is smaller than the distance between y_i and y_j themselves.

At the same time a path-based notion of separation between two nodes in a graph is the following.

Definition 14 (Path separation): Consider a graph $G = (V, \vec{E})$. We say that y_i and y_j are path separated by y_k if y_k is on every path from y_i to y_j.
It is quite straightforward to find examples where neighbor-separation defined by the distance (3) and path-separation do not match for the output processes $y_1, ..., y_n$ of a LDP.

Example 1: Consider the LDP

$$y_1 = e_1$$
$$y_2 = e_2$$
$$y_3 = y_1 + y_2 + e_3$$
$$y_4 = y_3 + e_4$$

where $\Phi_{e_1}(\mathbf{z}) = \Phi_{e_2}(\mathbf{z}) = \Phi_{e_3}(\mathbf{z}) = \Phi_{e_4}(\mathbf{z}) = 1$. The graphical representation of this LDP is represented in Figure 3 along with the distances between each pair of nodes. As shown in

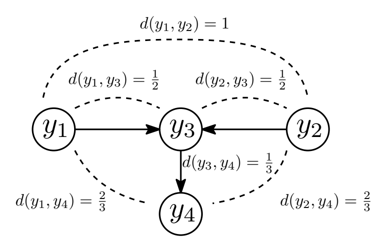

Fig. 3. A LDP where the notions of neighbor-separation and path-separation do not match.

Figure 3, we have $nsep(y_1|y_4|y_2)$, but y_4 is not on the path from y_1 to y_2.

However, a connection between the notion of neighbor-separation and path-separation can still be established.

Proposition 3.1: Consider a Linear Dynamic Influence Polytree $\mathcal{T} = (H(z), e)$ with outputs $(y_1, ..., y_n)$. We have that

- $\exists! y_k$ such that $nsep(y_i|y_k|y_j)$ implies y_k is the only element on the path from y_i and y_j
- $d(y_i, y_k) \leq d(y_\ell, y_k)$ for all $\ell \neq i, k$ and y_k is the only element on the path from y_i and y_j implies that $nsep(y_i|y_k|y_j)$.

Proof: From Theorem 8 in [25], the minimum spanning tree defined on the nodes according to distance (3) gives the graphical representation of the LDIM. Then the proof follows from the properties of minimum spanning trees: if y_i and y_j are not directly connected in the minimum spanning tree, if and only if the path connecting them contains at least a third node y_k. Since paths in minimum spanning trees are shortest paths, such a third node y_k is closer to both y_i and y_j more than y_i and y_j are close to each other. If y_k is unique, then it is the only node on the path from y_i to y_j. ■

Under certain conditions Proposition 3.1 provides an equivalence between neighbor-separation and path-separation in a LDP: if y_i is the closest element to y_k, then y_k is the only element such that $nsep(y_i|y_k|y_j)$ if and only if y_k is the only element that path-separates y_i and y_j. Thus, the results of Proposition 3.1 suggest an algorithm for the reconstruction of the skeleton of LDP from data: compute the coherence distance among the observed time series; test if there is exactly one node y_k such that $nsep(y_i|y_k|y_j)$ and in such a case conclude that $\{y_i, y_k\}$ and $\{y_j, y_k\}$ are in the skeleton of the LDP; apply this to all node pairs y_i,y_j.

`Neighbor Reconstruction Algorithm:`

---

1. Input: (V, d)
   where $V = \{y_1, ..., y_n\}$ is the ordered set of nodes and $d(y_i, y_j)$ is a distance between y_i and y_j.
2. Initialize $E = \emptyset$
3. For each pair $y_i, y_j \in V$
   - if $\exists! y_k$ such that $nsep(y_i|y_k|y_j)$
     add $\{y_i, y_k\}$ and $\{y_j, y_k\}$ to E
4. Output (V, E)

Proposition 3.2: Consider a Linear Dynamic Influence Polytree $\mathcal{T} = (H(z), e)$ with nodes $(y_1, ..., y_n)$ and define the coherence distance (3) on such nodes. Neighbor Reconstruction Algorithm outputs the skeleton of $\mathcal{T}$. Furthermore, the Neighbor Reconstruction Algorithm output is the Minimum Spanning Tree of the complete graph defined on the nodes $(y_1, ..., y_n)$ with weights given by (3).

Proof: The proof follows from Proposition 3.1. When y_k is the only node *nsep*-arating y_i and y_j according to the distance (3), then the edges y_i-y_k and y_j-y_k are in the undirected structure of the polytree. Once Step 3. is run on all pairs y_i and y_j, the algorithm outputs the full skeleton. ■

IV. Strictly causal networks

We can obtain strong results for the reconstruction of a LDIM when it is known that the dynamics on each edge is strictly causal.

Definition 15 (Strictly Causal LDIM): A LDIM $\mathcal{G} = (H(z), e)$ is strictly causal if each entry of $H(z)$ is strictly causal.

On a generic graph we define the notion of ancestor-separation. The following notion of separation is quite intuitive.

Definition 16 (ancestor-separation): Given a directed graph $G = (V, \vec{E})$, we say that the node y_i is ancestor-separated from the node y_j by a set y_Z if every path with no inverted forks from y_i to y_j has at least one element in Z. We denote this relation as $ancsep(y_i \to y_Z \to y_j)$.

Figure 4 illustrates ancestor-separation through few examples. Observe that, contrary to path-separation, ancestor-

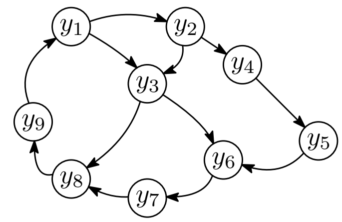

Fig. 4. Observe that $ancsep(y_1 \to \{y_3, y_7\} \to y_9)$ and $ancsep(y_6 \to y_9 \to y_2)$. Also, we have that y_3 does not ancestor-separate y_8 from y_1 because of the chain $y_1 \to y_2 \to y_4 \to y_5 \to y_6 \to y_7 \to y_8$.

separation is not symmetric since $ancsep(y_i \to y_Z \to y_j)$ does not necessarily imply $ancsep(y_j \to y_Z \to y_i)$.

ancestor-separation is a relation defined on the nodes of a graph. Also observe that there is no edge from y_i to y_j if and only if $ancsep(y_i \to y_Z \to y_j)$ for $Z = \{y_1, ..., y_n\} \setminus \{y_i, y_j\}$, namely y_j is ancestor-separated from y_i by all the other nodes.

In order to proceed according to the same approach followed in the previous section we need to define a probabilistic notion of separation among stochastic processes and obtain conditions under which the graphical notion and the probabilistic notions are equivalent.

The probabilistic notion that we introduce now is based on Granger-causality [26]. Granger causality is a widespread tool to infer the connectivity structure of a network of dynamic systems. Granger causality and derived methods test if past observations of a process y_i are useful in predicting the present observation of a process y_j, given full information about the past of all other processes. If such a test is positive, a causal relation from y_i to y_j is then identified, otherwise it is concluded that y_i does not cause y_j. Using the formalism provided by causal Wiener filters, a notion of separation that we refer to as Granger-separation, can be expressed as follows.

Definition 17 (Granger-separation): Given a vector of n wide sense stationary processes $y = (y_1, ..., y_n)$, we say that y_j is Granger-separated from y_i by y_Z, for $Z \subseteq \{1, ..., n\} \setminus \{i, j\}$, if the past of y_i, namely $\frac{1}{\mathbf{z}} y_i$, is not useful to estimate y_j when the past of y_Z, namely $\frac{1}{\mathbf{z}} y_Z$, and the past of y_j, namely $\frac{1}{\mathbf{z}} y_j$ are available. Using the Wiener filter formalism, this is equivalent to

$$W_{y_j \left[\frac{1}{z} y_i\right] | \frac{1}{z}(y_i, y_j, y_Z)} = 0,$$

and we denote this with $gsep(y_i \to y_Z \to y_j)$

In other words, Granger-separation means that, in order to make a one-step ahead prediction for y_j using the past of the signals y_Z, y_j and y_i, the information from the past of y_i is not needed. When considering $y_Z = \{y_1, ..., y_n\} \setminus \{y_i, y_j\}$, we immediately have that testing $gsep(y_i \to y_Z \to y_j)$ is equivalent to the standard Granger causality test.

Proposition 4.1: (*Equivalence between Granger causality and ancestor-separation)* Consider a strictly causal LDIM $\mathcal{G} = (H(z), e)$ with perfect representation $G = (V, \vec{E})$. We have that

$$ancsep_G(y_i \to y_Z \to y_j) \quad \Leftrightarrow \quad gsep(y_i \to y_Z \to y_j).$$

Proof: The proof makes use of the Wiener filter formalism for extending Granger causality in [27]. Specifically, the extension of Granger-causality of Theorem III.3 in [27] can be combined with Theorem III.2 in [28] after observing that if the set Z ancestor-separates y_i and y_j, then the Z is also meeting Definition 6 in [28]. The application of Theorem III.2 in [28] then gives the results. ■

Proposition 4.1 provides an immediate way to infer the graph underlying a strictly causal LDIM: compute the linear regression of y_j on the past of all signals, $\frac{1}{z}y$; and, if the coefficients associates with y_i are significantly different from zero, draw an edge from y_i to y_j. A similar result, that can provide also the spectral factor of the forcing input, is known in the literature of DSF [29]. However, the basic results is that Granger-causality tests return the graph of strictly causal LDIM suggesting the following algorithm.

Granger-causality Reconstruction Algorithm:

---

1. Input V where $V = \{y_1, ..., y_n\}$ is the ordered set of signals
2. Initialize $\vec{E} = \emptyset$.
3. For each pair $y_i, y_j \in V$
   - if $W_{y_j[\frac{1}{z}y_i]|\frac{1}{z}y} = 0$ then add (y_i, y_j) to $\vec{E}$
5. Output $(V, \vec{E})$

This algorithm is substantially equivalent to the reconstruction algorithm for DSF networks described in [29], where linear regressions in the time domain are used instead of Wiener filters.

V. Directed acyclic graphs

The results from the previous section consider very generic structures which can contain feedback loops as well. At the same time, they rely on relatively strong assumptions on the dynamics of the systems since each transfer function in the LDIM needs to be strictly causal. In this section we focus on results where, contrary to Section IV, the dynamics of the network is not required to be strictly causal and, contrary to Section III, the structure of the network is not as simple as a tree. Specifically, we consider LDIMs with graphical representations given by Directed Acyclic Graphs (DAGs), a widely adopted class of graphs in the theory of graphical models.

Definition 18 (Directed Acyclic Graphs): A directed graph $(V, \vec{E})$ is a Directed Acyclic Graph (DAG) if it has no directed loops.

The definition of a Directed Acyclic LDIM is straightforward.

Definition 19 (Directed Acyclic LDIM): A LDIM $\mathcal{G} = (H(z), e)$ is directed and acyclic if its graph representation is a DAG.

Following again the same approach considered in the two previous sections we will define a notion of separation on the graphical representation of the LDIM and a probabilistic notion of separation on the stochastic processes associated with the LDIM nodes. The graphical notion of separation in this case is slightly more complicated and requires the definition of forks and colliders on a path. The following definition introduces a notion of separation on subsets of vertices in a directed graph [20].

Definition 20 (d-separation and d-connection): Consider a directed graph $G = (V, \vec{E})$ and three mutually disjoint sets of vertices $y_I, y_Z, y_J \subseteq V$. The set y_Z is said to d-separate y_I and y_J if for every $y_i \in y_I$ and $y_j \in y_J$ all paths between y_i and y_j meet at least one of the following conditions:

1) the path contains a node $y_z \in y_Z$ that is not a collider
2) the path contains a collider y_k such that neither y_k nor its descendants belong to y_Z.

If y_Z d-separates y_I and y_J in the graph G, we write $dsep_G(y_I, y_Z, y_J)$. Otherwise we write $\neg dsep_G(y_I, y_Z, y_J)$ and say that y_Z d-connects y_I and y_J.

The notion of d-separation is one of the most fundamental concepts in graphical models. In Figure 5 we have used a simple graph to illustrate d-separation via few examples reported in the caption. An interesting property of d-separation

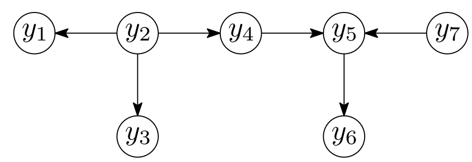

Fig. 5. Illustration of d-separation. The node x_2 is a fork on the path from x_1 to x_5 thus it blocks the path between x_1 and x_5, namely x_2 d-separates x_1 and x_5. The node x_4 is a chain link on the path from x_2 to x_6, thus it d-separates x_2 and x_6. The node x_5 is a collider on the path between x_3 and x_7. Since x_6 is a descendant of the collider x_5, we have that neither x_5 nor x_6 d-separate x_3 and x_7. However, the empty set d-separates x_3 and x_7 because of the collider x_5.

is that, in a DAG, two nodes are not directly connected by an edge if and only if there is a set y_Z that d-separates them.

Proposition 5.1: Let $\vec{G} = (V, \vec{E})$ be a DAG and let $G = (V, E)$ be its skeleton. We have that $\{y_i, y_j\} \notin E$ if and only if $\exists y_Z : dsep(y_i, y_Z, y_j)$.

Proof: The necessity is obtained by inspection by choosing Z as the union of set of parents of y_i and the set of parents of y_j. conversely we get the sufficiency by observing that, if there is an edge between y_i and y_j, the path associated with that edge cannot be d-separated by any set Z. ■

In the case of LDIMs with DAG structure, as a probabilistic notion of separation, we define Wiener-Hopf-separation.

Definition 21: (Wiener-separation) Given a vector of n wide sense stationary processes $y = (y_1, ..., y_n)$, we say that y_i and y_j are Wiener-Hopf-separated from y_i by y_Z, for $Z \subseteq \{1, ..., n\} \setminus \{i, j\}$, if y_i is not useful to estimate y_j when y_Z is available. Using the Wiener filter formalism, this is equivalent to

$$W_{y_j[y_i]|(y_i, y_Z)} = 0,$$

and we denote this with $whsep(y_i \to y_Z \to y_j)$.

We recall a result from [30] stating that d-separation in the DAG G underlying the LDIM implies Wiener-Hopf-separation.

Theorem 5.2 (Theorem 24 in [30]): Consider a LDIM $\mathcal{G} = (H(z), e)$ with graphical representation given by a

DAG $G = (V, \vec{E})$. We have that $dsep_G(y_i, y_Z, y_j)$ implies $wsep(y_i, y_Z, y_j)$.

Proof: See [30]. ■

The inverse implication (Wiener-Hopf-separation implies d-separation in a DAG) is not always true. When such implication holds we say that the DAG G is faithful to the LDIM.

Definition 22 (Faithfulness): We say that a LDIM is faithful to its graphical representation G if $wsep(y_i, y_Z, y_j)$ implies $dsep_G(y_i, y_Z, y_j)$.

As discussed in [31] and [32], faithfulness is a very mild assumption verified in virtually all practical applications. Indeed, consider a class of LDIMs with the same graphical representation where the transfer matrix $H(z)$ has each entry is parameterized according to the coefficients of of the polynomial at its numerator and its denominator, a basic results of [31] (see Theorem V.1 therein) is that a Lebesgue measure zero set of parameters give an unfaithful network. Thus, under the mild assumption of faithfulness, Wiener-Hopf-separation and d-separation are equivalent and the standard algorithm that exploits the equivalence between d-separation and a probabilistic notion of separation to reconstruct the skeleton of a graph is the Peter-Clark Algorithm invented by Peter Spirtes and Clark Glymour [33]. We report a variation of the Peter-Clark Algorithm for LDIMs.

`Peter-Clark Reconstruction Algorithm for LDIMs:`

---

1. Input V where $V = \{y_1, ..., y_n\}$ is the ordered set of signals
2. Initialize $E = V \times V$.
3. For each pair $y_i, y_j \in V$, with $y_i \neq y_j$
   - search for $y_Z \subseteq V \setminus \{y_i, y_j\}$ such that $dsep(y_i|y_Z|y_j)$
   - if y_Z exists, remove (y_i, y_j) from E
4. Output (V, E)

Conclusions

Various methods have been proposed in the literature to recover the network structure of Linear Dynamic Influence Models or similar mathematical descriptions, using additional a-priori knowledge about the network model, such as the topology being a tree or the dynamics being strictly causal. By considering a probabilistic notion of separation and a graphical notion of separation, the methods reviewed in this article have been reinterpreted in terms of a duality property. This duality property allows for the development of algorithms that can reconstruct the topology of particular network classes.

References

[1] V. Chetty, N. Woodbury, J. Brewer, K. Lee, and S. Warnick, "Applying a passive network reconstruction technique to twitter data in order to identify trend setters," in *Control Technology and Applications (CCTA), 2017 IEEE Conference on*. IEEE, 2017, pp. 1344–1349.

[2] M. Eisen, P. Spellman, P. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proc. Natl. Acad. Sci. USA*, vol. 95, no. 25, pp. 14 863–8, 1998.

[3] D. Del Vecchio, A. Ninfa, and E. Sontag, "Modular cell biology: Retroactivity and insulation," *Nature Molecular Systems Biology*, vol. 4, p. 161, 2008.

[4] P. E. Paré, C. L. Beck, and A. Nedich, "Epidemic processes over time-varying networks," *IEEE Transactions on Control of Network Systems*, 2017.

[5] A. Brovelli, M. Ding, A. Ledberg, Y. Chen, R. Nakamura, and S. L. Bressler, "Beta oscillations in a large-scale sensorimotor cortical network: directional influences revealed by Granger causality." *Proc Natl Acad Sci USA*, vol. 101, no. 26, pp. 9849–9854, June 2004.

[6] C. Quinn, T. Coleman, N. Kiyavash, and N. Hatsopoulos, "Estimating the directed information to infer causal relationships in ensemble neural spike train recordings," *Journal of computational neuroscience*, vol. 30, no. 1, pp. 17–44, 2011.

[7] J.-S. Bailly, P. Monestiez, and P. Lagacherie, "Modelling spatial variability along drainage networks with geostatistics," *Mathematical Geology*, vol. 38, no. 5, pp. 515–539, 2006.

[8] J. Gonçalves and S. Warnick, "Necessary and sufficient conditions for dynamical structure reconstruction of lti networks," *Automatic Control, IEEE Transactions on*, vol. 53, no. 7, pp. 1670–1674, 2008.

[9] K. P. Murphy, "Dynamic bayesian networks," *Probabilistic Graphical Models, M. Jordan*, vol. 7, 2002.

[10] P. M. Van den Hof, A. Dankers, P. S. Heuberger, and X. Bombois, "Identification of dynamic models in complex networks with prediction error methods—basic methods for consistent module estimates," *Automatica*, vol. 49, no. 10, pp. 2994–3006, 2013.

[11] S. J. Mason, "Feedback theory-some properties of signal flow graphs," *Proceedings of the IRE*, vol. 41, no. 9, pp. 1144–1156, 1953.

[12] N. Wiener, *Extrapolation, interpolation, and smoothing of stationary time series*. MIT press Cambridge, MA, 1949, vol. 2.

[13] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear estimation*. Prentice Hall, 2000, no. EPFL-BOOK-233814.

[14] S. Wright, "Correlation and causation," *Journal of agricultural research*, vol. 20, no. 7, pp. 557–585, 1921.

[15] ——, "The method of path coefficients," *The Annals of Mathematical Statistics*, vol. 5, no. 3, pp. 161–215, 1934.

[16] ——, "Path coefficients and path regressions: alternative or complementary concepts?" *Biometrics*, vol. 16, no. 2, pp. 189–202, 1960.

[17] A. S. Goldberger, "Structural equation methods in the social sciences," *Econometrica: Journal of the Econometric Society*, pp. 979–1001, 1972.

[18] K. A. Bollen and J. S. Long, *Testing structural equation models*. Sage, 1993, vol. 154.

[19] R. H. Hoyle, *Structural equation modeling: Concepts, issues, and applications*. Sage Publications, 1995.

[20] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[21] ——, *Causality*. Cambridge university press, 2009.

[22] D. Materassi and M. Salapaka, "On the problem of reconstructing an unknown topology via locality properties of the wiener filter," *Automatic Control, IEEE Transactions on*, vol. 57, no. 7, pp. 1765–1777, 2012.

[23] S. J. Mason, "Feedback theory: Further properties of signal flow graphs," 1956.

[24] D. Materassi, "Reconstruction of topologies for acyclic networks of dynamical systems," in *American Control Conference (ACC), 2011*. IEEE, 2011, pp. 37–41.

[25] F. Sepehr and D. Materassi, "Noninvasive approximation of linear dynamic system networks using polytrees," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 3, pp. 1314–1323, 2021.

[26] C. Granger, "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica*, vol. 37, pp. 424–438, 1969.

[27] M. Dimovska and D. Materassi, "A control theoretic look at granger causality: extending topology reconstruction to networks with direct feedthroughs," *IEEE Transactions on Automatic Control*, vol. 66, no. 2, pp. 699–713, 2020.

[28] S. Jahandari and D. Materassi, "Sufficient and necessary graphical conditions for miso identification in networks with observational data," *IEEE Transactions on Automatic Control*, vol. 67, no. 11, pp. 5932–5947, 2021.

[29] Z. Yue, J. Thunberg, W. Pan, L. Ljung, and J. Gonçalves, "Linear dynamic network reconstruction from heterogeneous datasets," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10 586–10 591, 2017.

[30] D. Materassi and M. V. Salapaka, "Signal selection for estimation and identification in networks of dynamic systems: a graphical model approach," *IEEE Transactions on Automatic Control*, vol. 65, no. 10, pp. 4138–4153, 2019.

[31] M. Dimovska and D. Materassi, "Granger-faithfulness and link orientation in network reconstruction," *IEEE Transactions on Control of Network Systems*, vol. 9, no. 1, pp. 113–124, 2022.

[32] J. Runge, V. Petoukhov, J. F. Donges, J. Hlinka, N. Jajcay, M. Vejmelka, D. Hartman, N. Marwan, M. Paluš, and J. Kurths, "Identifying causal gateways and mediators in complex spatio-temporal systems," *Nature communications*, vol. 6, p. 8502, 2015.

[33] P. Spirtes, C. N. Glymour, and R. Scheines, *Causation, prediction, and search*. MIT press, 2000.