Detection of Delays and Feedthroughs in Dynamic Networked Systems

Sina Jahandari, Ajitesh Srivastava

***Abstract*—The paper presents systematic tests to determine if a particular transfer function in an interconnected system is strictly proper or has a feedthrough. Considering a strictly proper module between two nodes in a networked system, if only the data of the two nodes are used, there are situations where numerical tests to determine if the module is strictly proper will fail. It is shown, however, that marginalizing some of the nodes of a networked system under certain conditions preserves delays and feedthroughs in certain links. Therefore, using a set of auxiliary nodes that satisfies certain conditions, it is possible to design a systematic test to determine if a module in the networked system is strictly proper. The conditions are proven to be sufficient. Similar ideas are used to formulate a systematic test to determine if a module has a feedthrough.**

***Index Terms*—Networked systems, Delays, Feedthroughs**

I. Introduction

An active line of research in studying interconnected systems is the identification of a particular module between two variables of a network. A main category of methods that attempt to address this problem are based on the prediction error method that predicts the output node using the information of a predictor inputs set containing the input node together with a set of auxiliary variables [1]–[6]. These techniques, however, assume that some a priori knowledge regarding the location of delays (strictly proper modules) and feedthroughs in the network is available. This paper deals with the problem of obtaining such information directly from data.

In a networked system, if a module of interest is strictly proper and only the data of the input node and the output node are used, there are situations where numerical tests will fail to determine that the module is strictly proper. This occurs, for example, when there is another parallel path between the two nodes containing modules with feedthroughs or when there is a confounding variable that influences both nodes through directed paths that contain modules with feedthroughs.

There has been some attempts in literature to test for the presence of delays and feedtroughs in networked systems directly from data [7]–[9]. While [7] proposes a two-test technique with the objective of detecting delays and feedthroughs, no theoretical guarantees are provided. On the other hand, the main goals of [8] and [9] are identification of the structure and dynamics of a networked system with direct feedthroughs.

Sina Jahandari is with the Electrical and Computer Engineering department, University of Southern California, Los Angeles, USA `jahandar@usc.edu`

Ajitesh Srivastava is with the Electrical and Computer Engineering department, University of Southern California, Los Angeles, USA `ajiteshs@usc.edu`

This article focuses on the problem of detecting from observational data whether a module embedded inside a networked system is strictly proper or has a direct feedthrough. A strictly proper module is a module where the degree of the numerator is less than the degree of the denominator. Considering a strictly proper module between two nodes in a networked system, if only the data of the two nodes are used, there are situations where numerical tests to determine if the module is strictly proper will fail. It is shown, however, that marginalizing some of the nodes of a networked system under certain conditions preserves delays and feedthroughs in certain links. Therefore, using a set of auxiliary nodes that satisfies certain conditions, it is possible to design a systematic test to determine if a module in the networked system is strictly proper. The conditions are proven to be sufficient. Similar ideas are used to formulate a systematic test to determine if a module has a feedthrough. The results could be seen as an extention and modification of the results of [8] and [9]. In particular, while the results of [8] and [9] treat delays and feedthroughs by attempting to obtain a ”recursive graphical representation” of the network for the objective of identification, this paper provides sharper theoretical results mainly focused on detection of strict causality of a module in the network directly from data. Moreover, the proposed conditions are less restrictive than the conditions required for consistent identification.

The article is organized as follows. Section II reviews some definitions and background information introducing networked systems and their associated graphs. In Section III, a motivating example is presented. Section IV presents the main result of the paper providing conditions under which marginalization of some of the nodes of a networked system preserves delays and feedthroughs in certain links. Section V presents a test for determining if a module has a delay. Section VI presents a test for determining if a module has a feedthrough. Concluding remarks are given in Section VII.

II. Preliminaries

In this section we briefly review some definitions and introduce our notation. We assume the reader is familiar with dynamic network identification literature and graph theory. The readers are referred to [4], [8], [10], [11] for more details.

A. A Model Class for Dynamic Networks

The class of networked systems considered in this paper is similar to the models studied in [3]–[5], [8], [9], [12].

Definition 1. *A networked system $\mathscr{G}$ is a pair $(H(z), n)$ where elements of the matrix $H(z)$ are causal discrete-time transfer functions, z is, in general, a complex number, and n is a vector of v mutually independent stochastic processes with rational power spectral density. The output signals $x_j(t)$ for $j = 1, ..., v$ are given by*

$$x_j(t) = n_j(t) + \sum_{i \in V} H_{ji}(z) x_i(t), \qquad \text{for } j = 1, ..., v \tag{1}$$

Note that the noise process $n_j(t)$ could be colored. Networked systems described by (1) could be represented via graphs. In a directed graph, a path between i and j is a sequence of distinct edges such that the first edge contains i, the last edge contains j and each two consecutive edges in the sequence are adjacent. Furthermore, if the edges have all the same orientation (as in $\{1 \to 3 \to 4 \to 5\}$) the path is called a dipath or a chain.

Definition 2. *Let $\mathscr{G} = (H, n)$ be a networked system with a set of nodes $V := \{1, ..., v\}$, and let E_1 be the set of single-headed edges and E_2 be the set of double-headed edges representing strictly proper modules such that*

(a) $i \to j \notin E_1 \cup E_2$ *implies* $H_{ji}(z) = 0$
(b) $i \to j \notin E_1$ *implies* $H_{ji}(z)$ *is strictly proper.*

We say that the multi-arrowed graph $G = (V, E_1, E_2)$ is a graph of the networked system [4].

Definition 3. *We say that the multi-arrowed graph $G = (V, E_1, E_2)$ is recursive if in every directed loop there is at least one double-headed edge.*

In a graph G, we say

- node j is a child of node i if $i \to j \in E_1 \cup E_2$ which is the same as node i being a parent of j. We denote the set of children of node j by $\mathrm{ch}_G(j) = \{v \in V | j \to v \in E_1 \cup E_2\}$ and the set of parents of i by $\mathrm{pa}_G(i) = \{v \in V | v \to i \in E_1 \cup E_2\}$.
- node j is a descendant of i if $j = i$ or if there is a dipath from i to j. In such a case, node i is an ancestor of node j. The set of descendants of i is denoted by $\mathrm{de}_G(i)$ and the set of ancestors of i is denoted by $\mathrm{an}_G(i)$.

Definition 4. *Given a graph $G = (V, E_1, E_2)$, the graph of instantaneous propagations $G^{\not{t}}$ is the subgraph obtained by removing all the double-headed edges of G.*

Definition 5. *A node j in a path π in a graph G is a*

- *a fork, when π is of the form $i \leftarrow j \to k$*
- *collider, when π is of the form $i \to j \leftarrow k$*
- *a chain link, when π is of the form $i \to j \to k$*

Definition 6. *Given a graph G, we say that a set of nodes Z block the path π if*

- *there exists a non-collider i on π such that $i \in Z$; or*
- *there exists a collider c on π such that $de_G(c) \cap Z = \emptyset$.*

Definition 7. *We say that a path π is j-pointing if the last edge of π is of the form $k \to j$ for some node k.*

Definition 8. *The natural filtration generated by a set of stochastic processes x_A, up to time t is denoted by $I_A(t)$.*

By the notation of Definition 8 the estimate $\hat{x}_j(t)$ of $x_j(t)$ in the least square sense based on the information of variables x_{D^+} up to time t and the information of variables x_{D^-} up to time $t-1$ could be written as

$$\hat{x}_j(t) = \mathbb{E}(x_j(t) \mid I_{D^+}(t), I_{D^-}(t-1)). \tag{2}$$

where $W_{jk}(z)$ for $k \in D^+$ are proper modules and for $k \in D^-$ are strictly proper modules. In the linear Gaussian case Equation (2) reduces to to

$$\hat{x}_j = \sum_{k \in D^+} W_{jk}(z) x_k + \sum_{k \in D^-} W_{jk}(z) x_k. \tag{3}$$

B. Marginalization in dynamic networks

Here, we explain the notion of marginalizing [13] certain nodes of the the networked system. The marginalization of a node in a networked system is the process of defining a new system with the same variables, but such that the incoming links in the marginalized node are removed [13]. For example, consider the network of Figure 1 where node 4 is not measured. For the process $x_2(t)$ we can write

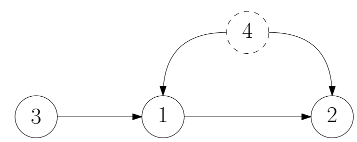

Fig. 1. Graphical representation of the network discussed in Section II-B

$$\begin{aligned} x_2(t) &= n_2(t) + H_{24}(z)x_4(t) + H_{21}(z)x_1(t) \\ &= n_2(t) + H_{21}(z)(n_1(t) + H_{13}(z)x_3(t) + H_{14}(z)x_4(t)) + H_{24}(z)x_4 \\ &= n_2(t) + \big(H_{24}(z) + H_{21}(z)H_{14}(z)\big)x_4(t) \\ &\quad + H_{21}(z)n_1(t) + H_{21}(z)H_{13}(z)x_3(t) \end{aligned} \tag{4}$$

If we marginalize nodes 3 and 4 and reduce the network

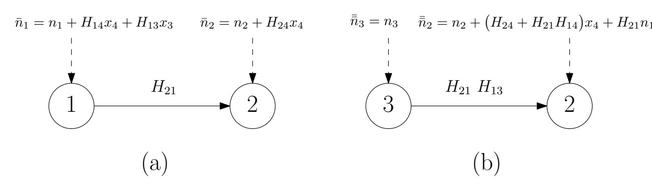

Fig. 2. (a) Marginalizing nodes 3 and 4 (b) Marginalizing nodes 1 and 4

to nodes 1 and 2 we will have a network with a graphical representation $\bar{G}$ of Figure 2 (a). In this new network we have $\bar{x}_1(t) = x_1(t)$, $\bar{x}_2(t) = x_2(t)$, $\bar{H}_{21}(z) = H_{21}(z)$,

$$\bar{n}_1(t) = n_1(t) + H_{14}(z)x_4(t)(t) + H_{13}(z)x_3(t), \tag{5}$$

, and $\bar{n}_2(t) = n_2(t) + H_{24}(z)x_4(t)$. In the following we explain further why the unmeasured confounding variable node 4 does not hinder the identification of H_{23} in the same way it does for H_{21}. Since $\bar{n}_1(t)$ and $\bar{n}_2(t)$ are correlated (because of the terms associated with $x_4(t)$) when we use the information of $x_1(t)$ to estimate $x_2(t)$, we get a biased estimate for $\bar{H}_{21}(z)$. However, if we marginalize nodes 1 and 4 and reduce the network to nodes 2 and 3 we will have a network with a graphical representation $\bar{\bar{G}}$ of Figure 2 (b). In this new network we have $\bar{\bar{x}}_2(t) = x_2(t)$, $\bar{\bar{x}}_3(t) = x_3(t)$, $\bar{\bar{H}}_{23}(z) = H_{21}(z)\ H_{13}(z)$,

$$\bar{\bar{n}}_2(t) = n_2(t) + \big(H_{24}(z) + H_{21}(z)H_{14}(z)\big)x_4(t) + H_{21}(z)n_1(t) \tag{6}$$

and $\bar{\bar{n}}_3(t) = n_3(t)$. Since $\bar{\bar{n}}_2(t)$ and $\bar{\bar{n}}_3(t)$ are uncorrelated when we use the information of $x_3(t)$ to estimate $x_2(t)$, we get a consistent estimate for $\bar{\bar{H}}_{23}(z)$ which is equivalent to $H_{21}(z)\ H_{13}(z)$.

III. Motivating example

Consider a simple two-node network with a block diagram depicted in Figure 3 (a). A recursive graph of the network is depicted in Figure 3 (b). Suppose the objective is the identification of module $H_{ji}(z)$. Note that nodes i and j are involved in a feedback loop. Since the edge from i to j is double-headed, we know that the module $H_{ji}(z)$ is strictly proper. This information is crucial for the identification. Indeed, considering a standard prediction error method, the quantity $\hat{H}_{ji} = \frac{W_{ji}(z)}{1-W_{jj}(z)}$ where $W_{ji}(z)$ and $W_{jj}(z)$ are computed from

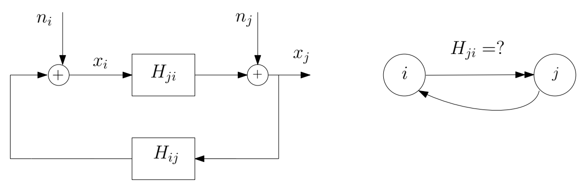

Fig. 3. (a) Block diagram of a two-node networked system (b) A recursive graph of the networked system containing two nodes in a feedback loop

$$\mathbb{E}(x_j(t)|I_j(t-1), I_i(t)) = W_{jj}x_j + W_{ji}x_i.$$

when we include the information of x_i up to time t, is a biased estimate of $H_{ji}(z)$. However, if we include the information of x_i up to time $t-1$

$$\mathbb{E}(x_j(t)|I_{i,j}(t-1)) = W_{jj}x_j + W_{ji}x_i.$$

the quantity $\hat{H}_{ji} = \frac{W_{ji}(z)}{1-W_{jj}(z)}$ is a consistent estimate of $H_{ji}(z)$.

The above example shows one instance of why it is important to be able to obtain information about delays and feedthroughs in modules embeded in a networked system when such information is not available.

For a simple two node networked system described by $x_j(t) = H_{ji}(z)x_i(t) + n_j(t)$, it is possible to use the information of variables x_i and x_j to consistently identify the module $H_{ji}(z)$ and, consequently, observe if it is has a delay or not. However, if the networked system is more complex, the information of variables x_i and x_j would not, in general, be sufficient to consistently identify $H_{ji}(z)$ or to determine if it has a delay. Many works [2], [9], [13]–[15] have tried to provide sufficient conditions for a set of additional auxiliary variables to guarantee consistent estimation of a particular module in a dynamic networked system. All these techniques, however, require a priori information about the location of delays and feedthroughs. Indeed, it is crucial to know whether to include in the estimator the information of a certain auxiliary variable up to time t or up to time $t-1$, as was shown in the motivating example of Section III. Similarly, when the objective is merely determining if a particular module $H_{ji}(z)$ has a delay or not, using the information of variables x_i and x_j would not, in general, be sufficient. For example, if $H_{ji}(z)$ has a delay and there is another parallel path between the two nodes i and j containing modules with feedthroughs or when there is a confounding variable that influences both nodes i and j through directed paths that contain modules with feedthroughs, the information of x_i and x_j would not be sufficient to determine whether $H_{ji}(z)$ has a delay.

IV. Main result

This section presents the main result of the paper providing conditions to investigate whether a certain module in a networked system has a delay or not. Although blocking all the j-pointing paths between two nodes i and j does not guarantee the consistent identification of the module $H_{ji}(z)$ in a networked system, the following theorem shows it suffices to analyze whether $H_{ji}(z)$ has a delay or not (for consistent identification we need more conditions, see [3], [4]). This is done through the analysis of marginalizing [13] certain nodes of the the networked system. The marginalization of a node in a networked system is the process of defining a new system with the same variables, but such that the incoming links in the marginalized node are removed [13].

Theorem IV.1. *Consider a networked system $(H(z), n)$ with no algebraic loops and with graphical representation $G = (V, E_1, E_2)$. Assume the set Z blocks all the j-pointing paths between i and j with the exception of $i \to j$ in G. Let Q be a set of nodes such that $Q \cap (Z \cup \{i, j\}) = \emptyset$. Let $G^r = (V, E_1^r, E_2^r)$ be a graph obtained from G in the following way.*

1) *if there is at least one dipath from k to $\ell \in V \setminus Q$ in G with all internal nodes in Q and all single-headed edges, then $k \to \ell \in V \setminus Q$ is in E_1^r*
2) *if all the dipaths from k to ℓ in G with all internal nodes in Q have at least one double-headed edge, then $k \to \ell$ is in E_2^r*
3) *if there is no dipath from k to ℓ in G with all internal nodes in Q, or $\ell \in Q$, then $k \to \ell \notin E_1^r \cup E_2^r$*

We have that

- *G^r is a graphical representation of the network $(H^r(z), n)$ obtained by marginalizing Q, (see Lemma 15 in [13])*
- *$Z \cap de_{G^{\not\,\iota}}(j) = Z \cap de_{G^{r\not\,\iota}}(j)$*
- *$Z \cap an_{G^{\not\,\iota}}(j) = Z \cap an_{G^{r\not\,\iota}}(j)$*
- *$H_{ji}^r(z)$ is strictly proper if and only if $H_{ji}(z)$ is strictly proper*
- *all j-pointing paths between i and j except $i \to j$ are blocked by Z in G^r*

Proof. See the appendix. □

One of the main points that Theorem IV.1 shows is that when a set of nodes Z blocks all the j-pointing paths between nodes i and j with the exception of $i \to j$ in a graph of a networked system, if we reduce the networked system to i, j, and Z by marginalization, the module between i and j in the new system has a delay (or feedthough) if and only if the module between i and j in the original system had a delay (or feedthough).

V. Detection of delays in networked systems

The following result, however, provides a test that using a set of auxiliary variables determines if a particular module has a delay. The sufficient conditions that the set of auxiliary variables needs to satisfy are less restrictive than the conditions required for consistent identification.

Theorem V.1. *Consider a set of nodes A that blocks all the j-pointing paths between nodes i and j with the exception of $i \to j$ in the graph G of a networked system. If the Wiener filter $W_{ji}(z)$ in*

$$\mathbb{E}(x_j(t) \mid I_{\{j\}\cup A^-}(t-1), I_{A^+\cup\{i\}}(t)) = \sum_{r\in A^-\cup A^+\cup\{i,j\}} W_{jr}(z)x_r(t) \tag{7}$$

had a delay for all possible combinations of disjoint A^- and A^+ with $A^-\cup A^+ = A$, then $H_{ji}(z)$ would have a delay.

Proof. See the appendix. □

The following example shows how Theorem V.1 can be applied to determine if a module in a networked system has a delay.

Example 1. *Consider a networked system with an unknown true graph depicted in Figure 4 (a). It is assumed that node 5 which is depicted with dashed lines is not measured. It can be seen from Figure 4 (a) that modules $H_{32}(z)$ and $H_{43}(z)$ have delays. This available graph of the network is depicted in Figure 4 (b). As can be seen in Figure 4 (b), the information that modules $H_{32}(z)$ and $H_{43}(z)$ have delays is not available. Suppose the objective is to obtain this information directly from data of the nodes of the network, namely, to determine if the modules $H_{32}(z)$ and $H_{42}(z)$ have delays from $I_{1,2,3,4,5}(t)$.*

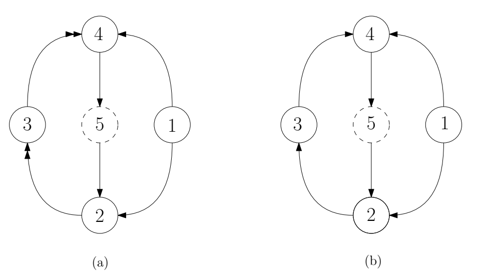

Fig. 4. (a) The true recursive graph of the networked system of Example 1 which is not known. It is assumed that node 5 which is depicted with dashed lines is not measured.; (b) The available graph of the networked system of Example 1. The goal is to determine whether the modules $H_{32}(z)$ and $H_{42}(z)$ have delays or not.

Applying Theorem V.1 on the module $H_{32}(z)$ we can see that there is no 3-pointing paths between nodes 2 and 3 except $2 \to 3$. That is, the set $A = \{\emptyset\}$. Since the Wiener filter$W_{32}(z)$ in

$$\mathbb{E}(x_3(t) \mid I_3(t-1), I_2(t) = W_{32}(z)x_2(t) + W_{33}(z)x_2(t) \tag{8}$$

is strictly proper, it follows from Theorem V.1 that $H_{32}(z)$ has a delay. On the other hand, Applying Theorem V.1 on the module $H_{43}(z)$ we can see that the set $A = \{2\}$ blocks all the 4-pointing paths between the nodes 3 and 4, namely, $\{3 \leftarrow 2 \leftarrow 1 \rightarrow 4\}$. By Theorem V.1 we have to consider two cases for A^+ and A^-. The first possible combination is $A^+ = \{2\}$ and $A^- = \{\emptyset\}$. In this case, the Wiener filter $W_{43}(z)$ in

$$\mathbb{E}(x_4(t) \mid I_4(t-1), I_{2,3}(t) = \sum_{r\in\{2,3,4\}} W_{4r}(z)x_r(t) \tag{9}$$

turns out to have a delay. The second possible combination is $A^+ = \{\emptyset\}$ and $A^- = \{2\}$. In this case, the Wiener filter $W_{43}(z)$ in

$$\mathbb{E}(x_4(t) \mid I_{2,4}(t-1), I_3(t) = \sum_{r\in\{2,3,4\}} W_{4r}(z)x_r(t) \tag{10}$$

also turns out to have a delay. Therefore, it follows from Theorem V.1 that $H_{32}(z)$ has a delay.

VI. Detection of feedthroughs in networked systems

In the previous result we saw how we can infer if a module has a delay. The following result, on the other hand, provides a test to determine if a module has a nonzero feedthrough.

Theorem VI.1. *Consider a set of nodes A that blocks all the i-pointing and j-pointing paths between nodes i and j with the exception of $i \to j$ and $j \to i$ in the graph $G = (V, E_1, E_2)$ of a networked system. If the Wiener filter $W_{ji}(z)$ in*

$$\mathbb{E}(x_j(t) \mid I_{\{j\}\cup A^-}(t-1), I_{A^+\cup\{i\}}(t)) = \sum_{r\in A^-\cup A^+\cup\{i,j\}} W_{jr}(z)x_r(t) \tag{11}$$

and the Wiener filter $W_{ij}(z)$ in

$$\mathbb{E}(x_i(t) \mid I_{\{i\}\cup A^-}(t-1), I_{A^+\cup\{j\}}(t)) = \sum_{\ell\in A^-\cup A^+\cup\{i,j\}} W_{ir}(z)x_\ell(t) \tag{12}$$

have a feedthrough for all possible combinations of disjoint A^- and A^+ with $A^-\cup A^+ = A$, then either the transfer function $H_{ji}(z)$ has a feedthrough or the transfer function $H_{ij}(z)$ has a feedthrough.

Proof. See the appendix. □

The following example shows how Theorem VI.1 can be applied to determine if a module in a networked system has a feedthrough.

Example 2. *Consider a networked system with an unknown graph depicted in Figure 5 (a). What is available, however, is the graph of Figure 5 (b). It is known that the networked system does not have any algebraic loops. The objective is to determine whether the module $H_{23}(z)$ has a feedthough or not. It can be seen that, the set $A = \{1\}$ blocks all the 2-pointing and 3-pointing paths between the nodes 2 and 3. Following Theorem VI.1, we consider two possible combinations of A^+ and A^-. In the first combination, we consider $A^+ = \{1\}$ and $A^- = \{\emptyset\}$. In this case, the Wiener filter $W_{23}(z)$ in*

$$\mathbb{E}(x_2(t) \mid I_2(t-1), I_{1,3}(t)) = \sum_{r\in\{1,2,3\}} W_{2r}(z)x_r(t) \tag{13}$$

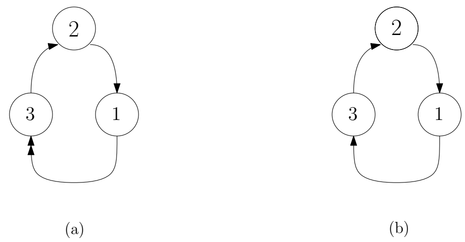

Fig. 5. (a) The true recursive graph of the networked system of Example 2 which is not known; (b) The available graph of the networked system of Example 2. The goal is to determine whether the module $H_{23}(z)$ has a feedthrough.

and $W_{32}(z)$ in

$$\mathbb{E}(x_3(t) \mid I_3(t-1), I_{1,2}(t)) = \sum_{r \in \{1,2,3\}} W_{3r}(z) x_r(t) \tag{14}$$

turn out to have feedthroughs. In the second combination, we consider $A^+ = \{\emptyset\}$ and $A^- = \{1\}$. In this case, the Wiener filter $W_{23}(z)$ in

$$\mathbb{E}(x_2(t) \mid I_{2,1}(t-1), I_3(t)) = \sum_{r \in \{1,2,3\}} W_{2r}(z) x_r(t) \tag{15}$$

and $W_{32}(z)$ in

$$\mathbb{E}(x_3(t) \mid I_{1,3}(t-1), I_{2,}(t)) = \sum_{r \in \{1,2,3\}} W_{3r}(z) x_r(t) \tag{16}$$

also turn out to have feedthroughs. Therefore, based on Theorem VI.1 we can conclude that $H_{23}(z)$ has a feedthrough.

VII. Conclusion

The paper dealt with the problem of determining whether a particular module in an interconnected system has a delay or a feedthrough. It was shown that when marginalizing some of the nodes of the network and reducing the network to the target nodes and a set of auxiliary variables that satisfies certain conditions, delays and feedthroughs will be preserved on the particular module connecting the target nodes. This enabled designing a statistical test that determined if a particular module in the networked system had a delay. Another test which required more complicated conditions was developed to determine if a module has a feedthrough. The provided sufficient conditions are less restrictive compared to conditions required for consistent identification.

References

[1] A. Dankers, P. M. Van den Hof, X. Bombois, and P. S. Heuberger, "Errors-in-variables identification in dynamic networks—consistency results for an instrumental variable approach," *Automatica*, vol. 62, pp. 39–50, 2015.

[2] K. R. Ramaswamy and P. M. Vandenhof, "A local direct method for module identification in dynamic networks with correlated noise," *IEEE Transactions on Automatic Control*, 2020.

[3] S. Jahandari and D. Materassi, "Optimal observations for identification of a single transfer function in acyclic networks," *IEEE Conference on Decision and Control*, 2021.

[4] ——, "Optimal selection of observations for identification of multiple modules in dynamic networks," *conditionally accepted in IEEE Transaction on Automatic Control*, 2022.

[5] S. Jahandari, A. Kalhor, and B. N. Araabi, "Online forecasting of synchronous time series based on evolving linear models," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 5, pp. 1865–1876, 2018.

[6] ——, "Order determination and robust adaptive control of unknown deterministic input-affine systems: An operational controller," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 3831–3836.

[7] 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, no. 1, pp. 1–10, 2015.

[8] 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.

[9] S. Jahandari and D. Materassi, "Sufficient and necessary graphical conditions for miso identification in networks with observational data," *IEEE Transaction on Automatic Control*, 2021.

[10] A. Dankers, P. M. Van den Hof, X. Bombois, and P. S. Heuberger, "Identification of dynamic models in complex networks with prediction error methods: Predictor input selection," *IEEE Transactions on Automatic Control*, vol. 61, no. 4, pp. 937–952, 2015.

[11] J. Pearl, *Causality: models, reasoning and inference*. Springer, 2000, vol. 29.

[12] S. Jahandari and D. Materassi, "Identification of dynamical strictly causal networks," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 4739–4744.

[13] 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*, 2019.

[14] A. G. Dankers, P. M. Van den Hof, P. S. Heuberger, and X. Bombois, "Dynamic network structure identification with prediction error methods-basic examples," *IFAC Proceedings Volumes*, vol. 45, no. 16, pp. 876–881, 2012.

[15] A. Dankers, P. M. Van den Hof, D. Materassi, and H. H. Weerts, "Conditions for handling confounding variables in dynamic networks," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3983–3988, 2017.

Appendix

A. Proof of Theorem IV.1

Proof. Following the proof of property 2) of Lemma 15 in [13] (Node Marginalization Lemma), we find that the graph G^r is a graphical representation of the reduced model. In the graph G^r, the nodes in Q are not descendants of any node since they have no incoming edges. Furthermore, we have that $\mathrm{de}_{G^r}(k) = \mathrm{de}_G(k) \setminus Q$ for all $k \notin Q$. Indeed, if there is a dipath π in G from k to $\ell \notin Q$, replace every sequence in π of the form $v \to a_1 \to \ldots \to a_M \to w$ where $\{a_1, \ldots, a_M\} \subseteq Q$ with $v \to w \in E^r$ to obtain the dipath π^r. We have that π^r is a dipath in G^r from k to ℓ. To establish that $Z \cap \mathrm{de}_{G^\sharp}(j) = Z \cap \mathrm{de}_{G^{r\sharp}}(j)$ we show that $Z \cap \mathrm{de}_{G^\sharp}(j) \subseteq Z \cap \mathrm{de}_{G^{r\sharp}}(j)$ and $Z \cap \mathrm{de}_{G^\sharp}(j) \supseteq Z \cap \mathrm{de}_{G^{r\sharp}}(j)$. Suppose $y \in Z \cap \mathrm{de}_{G^\sharp}(j)$. That is, there exists a dipath π with all single-headed edges from j to y in G. If no internal node on π is in Q, then the very same path exists in G^r. If some of the nodes on π are also in Q, then π has the form $j \cdots k \to a_1 \to \cdots a_m \to \ell \cdots \to y$, where k and ℓ are not in Q and $a_1 \cdots a_m \in Q$. Then by condition 1) of the Lemma, there exists a single-headed edge from k to ℓ in G^r that can be used to replace $k \to a_1 \to \cdots a_m \to \ell$ to $k \to \ell$. We can iterate this procedure to eliminate all internal nodes in Q. Eventually, we find a path $\pi' = j \cdots k \to \ell \cdots \to y$ with all single-headed edges from j to y in G^r. Therefore, we have that $y \in Z \cap \mathrm{de}_{G^{r\sharp}}(j)$, giving $Z \cap \mathrm{de}_{G^\sharp}(j) \subseteq Z \cap \mathrm{de}_{G^{r\sharp}}(j)$. Now suppose, $y \in Z \cap \mathrm{de}_{G^{r\sharp}}(j)$. That is, there exists a dipath with all single-headed edges from j to y in G^r. Then, it follows from condition 1) of the Lemma that there exists a dipath with all single-headed edges from j to y in G. Therefore, we have that $y \in Z \cap \mathrm{de}_{G^{\not\,}}(j)$, giving $Z \cap \mathrm{de}_{G^{\not\,}}(j) \supseteq Z \cap \mathrm{de}_{G^{r\not\,}}(j)$. The assertion $Z \cap \mathrm{an}_{G^{\not\,}}(j) = Z \cap \mathrm{an}_{G^{r\not\,}}(j)$ can be established using analogous steps swapping the roles of j and y. We prove that all j-pointing paths between i and j except $i \to j$ are blocked by Z in G^r by contradiction. By contradiction assume that there exists a path π^r between i and j that is not blocked by Z in G^r. Construct a new path π from π^r in the following way: for all edges $k \to \ell$ in π^r such that $k \to \ell \notin E$, replace $k \to \ell$ with sequence of edges $k \to a_1 \to ... \to a_M \to \ell$ with $\{a_1,...,a_M\} \subseteq Q$. Such a sequence of edges $k \to a_1 \to ... \to a_M \to \ell$ is always a dipath in G because of the way G^r has been constructed. Observe that π is a path in G. Furthermore, π and π^r have the same colliders. We now want to show that π is not blocked by Z in G. If π^r has no colliders, then π has no colliders either. Hence, since π^r is not blocked by Z, π is not blocked by Z leading to a contradiction. Consider, then, the case where π^r has all active colliders, and no non-colliders in Z. Because of the way π was obtained, π has no non-colliders in Z. Also, a collider c in π^r is a collider in π. Because $\mathrm{de}_{G^r}(c) = \mathrm{de}_G(c) \setminus Q$ and $Z \cap Q = \emptyset$ we have $\mathrm{de}_{G^r}(c) \cap Z = \mathrm{de}_G(c) \cap Z$. Hence a collider on π^r activated by Z in G^r is also a collider on π activated by Z in G. This again leads to a contradiction. □

B. *Proof of Theorem V.1*

Proof. First we consider the scenario where all j-loops are blocked by A. Let $G^p = (V, E_1^p, E_2^p)$ be the perfect graphical representation of the network. Since the network has no algebraic loops, G^p is recursive. Build a new graphical representation $\overline{G} = (V, \overline{E}_1, \overline{E}_2)$ of the network by adding single-headed edges from all nodes $k \in Z^+$ to j in G^p. That is, $\overline{E}_2 = E_2^p$ and $\overline{E}_1 = E_1^p \cup_{k \in Z^+} \{k \to j\}$. This implies that $Z^+ \subseteq \mathrm{an}_{\overline{G}^{\not\,}}(j)$. Note that $\overline{G}$ is recursive because for all edges $k \to j$ that we added to E_1^p to obtain $\overline{E}_1$, we have that $k \notin \mathrm{de}_{\overline{G}^{\not\,}}(j)$. Assume, by contradiction that $H_{ji}(z)$ is not strictly proper. Define $Z^- := \{\ell \in A : \ell \notin \mathrm{an}_{G^{\not\,}}(j)\}$, $Z^+ := \{k \in A : k \notin \mathrm{de}_{G^{\not\,}}(j)\} \setminus Z^-$, and $A := A \setminus (Z^- \cup Z^+)$. Since $Z^- := \{\ell \in A : \ell \notin \mathrm{an}_{G^{\not\,}}(j)\}$, we have that $Z^- \cap \mathrm{an}_{G^{p\not\,}}(j) = \emptyset$. Since Z^- does not contain any ancestor of Z^+ in $G^{\not\,}$, it also follows that $Z^- \cap \mathrm{an}_{\overline{G}^{\not\,}}(j) = \emptyset$. Hence, applying Theorem III.2 of [9] on $\overline{G}$, we get that $Z^- \subset D^-$. On the other hand, since $Z^+ \subseteq \mathrm{an}_{\overline{G}^{\not\,}}(j)$, we have that $Z^+ \subset D^+$. Since i is a parent of j in $\overline{G}^{\not\,}$, which is a recursive graph, there is one choice of A^- and A^+ where $D^- = Z^- \cup A^- \cup \{j\}$ and $D^+ = Z^+ \cup A^+ \cup \{i\}$ meeting the conditions of Theorem III.2 of [9] on $\overline{G}$. For those A^- and A^+ Theorem III.2 of [9] gives

$$W_{ji}(z) = (1 - W_{jj}(z))^{-1} H_{ji}(z). \tag{17}$$

Since $W_{jj}(z)$ is strictly proper, we necessarily have that $H_{ji}(z)$ is strictly proper which is a contradiction. If A does not block all j-loops, then marginalize the network $(H(z), n)$ with respect to the nodes $V \setminus (A \cup \{i, j\})$ and obtain the reduced network $(H^r(z), n^r)$ as in Theorem IV.1. Since the original network $(H(z), n)$ has no algebraic loops, the reduced network has no algebraic loops either. Again because of Theorem IV.1, all j-pointing paths between i and j that are not the edge $i \to j$ are blocked by A in G^r. Furthermore, since the only nodes in G^r are $A \cup \{i, j\}$, all j-loops are blocked by $A \cup \{i\}$. Hence, we can apply the same argument to the reduced network (H^r, n^r) and conclude that $H_{ji}^r(z)$ is strictly proper. Again, because of Theorem IV.1, the module $H_{ji}(z)$ is going to be strictly proper if and only if $H_{ji}^r(z)$ is strictly proper proving the assertion. □

C. *Proof of Theorem VI.1*

Proof. Without any loss of generality, assume that $\{j \to i\} \in E_1$, otherwise we can redefine E_1 and E_2 respectively as $E_1 \cup \{j \to i\}$, and $E_2 \setminus \{j \to i\}$, since this would still give us a (non-necessarily recursive) graphical representation of the network. Let $G^p = (V, E_1^p, E_2^p)$ be the perfect graphical representation of the network. Since the network has no algebraic loops, G^p is recursive. Since G^p is recursive it holds that (i) every dipath from j to i has at least a double headed edge or (ii) every dipath from i to j has at least a double headed edge. Consider first case (i). As in the proof of Theorem V.1, we first assume that $A \cup \{i\}$ blocks all j-loops. Define $Z^- := \{\ell \in A : \ell \notin \mathrm{an}_{G^{\not\,}}(j)\}$, $Z^+ := \{k \in A : k \notin \mathrm{de}_{G^{\not\,}}(j)\} \setminus Z^-$, and $A := A \setminus (Z^- \cup Z^+)$. Then, build a graphical representation $\overline{G} = (V, \overline{E}_1, \overline{E}_2)$ of the network by adding the single headed edge $i \to j$ and single-headed edges from all nodes $k \in Z^+$ to j in G^p. That is, $\overline{E}_2 = E_2^p$ and $\overline{E}_1 = E_1^p \cup \{i \to j\} \cup_{k \in Z^+} \{k \to j\}$. This implies that $Z^+ \cup \{i\} \subseteq \mathrm{an}_{\overline{G}^{\not\,}}(j)$. Since Z^- does not contain any ancestor of Z^+ in $G^{\not\,}$, it also follows that $Z^- \cap \mathrm{an}_{\overline{G}^{\not\,}}(j) = \emptyset$. Observe also that because of (i) and the definition of Z^+, $\overline{G}$ is recursive. Hence, by applying Theorem III.2 of [9] on $\overline{G}$, we get that there exist disjoint A^- and A^+ such that $D^- = A^- \cup Z^- \cup \{j\}$ and $D^+ = A^+ \cup Z^+ \cup \{i\}$ giving a consistent estimate of the module $H_{ji}(z)$. Since for all choices of A^- and A^+ the module estimate that would result from Equation (11) has a non-zero feedthrough component, $H_{ji}(z)$ needs to be non-strictly proper under scenario (i) when all j-loops are blocked by $A \cup \{i\}$. If $A \cup \{i\}$ does not block all j-loops then marginalize the network $(H(z), n)$ with respect to the nodes $A = V \setminus (A \cup \{i, j\})$ and obtain the reduced network $(H^r(z), n^r)$ as in Theorem IV.1. Since the original network $(H(z), n)$ has no algebraic loops, the reduced network has no algebraic loops either. Again because of Theorem IV.1, all j-pointing paths between i and j that are not the edge $i \to j$ are blocked by A in G^r. Furthermore, since the only nodes in G^r are $A \cup \{i, j\}$, all j-loops are blocked by $A \cup \{i\}$. Hence we can apply the same argument to the reduced network (H^r, n^r) and conclude that $H_{ji}^r(z)$ is not strictly proper. Again, because of Theorem IV.1, the module $H_{ji}(z)$ is going to be strictly proper if and only if $H_{ji}^r(z)$ is strictly proper proving the assertion. If instead scenario (ii) holds, we build a graphical representation $\overline{G} = (V, \overline{E}_1, \overline{E}_2)$ of the network by adding the single headed edge $j \to i$ and single-headed edges from all nodes $k \in Z^+$ to i in G^p. By repeating steps similar to scenario (i) with reversed roles for the nodes i and j, we would find that, in scenario (ii), because of Equation (12) the module $H_{ij}(z)$ needs to be non-strictly proper. Now, we do not know if scenario (i) or scenario (ii) holds, thus, we can only conclude that either $H_{ji}(z)$ is strictly proper or $H_{ij}(z)$ is strictly proper. □