



# Abstractions of linear dynamic networks for input selection in local module identification<sup>☆</sup>

Harm H.M. Weerts<sup>a</sup>, Jonas Linder<sup>b</sup>, Martin Enqvist<sup>c</sup>, Paul M.J. Van den Hof<sup>a,\*</sup>

<sup>a</sup> Control Systems Group, Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands

<sup>b</sup> ABB Corporate Research, Västerås, Sweden

<sup>c</sup> Division of Automatic Control, Linköping University, Sweden

## ARTICLE INFO

### Article history:

Received 2 January 2019  
Received in revised form 12 September 2019  
Accepted 16 March 2020  
Available online xxxx

### Keywords:

Dynamic networks  
System identification  
Closed-loop identification  
Graph theory

## ABSTRACT

In abstractions of linear dynamic networks, selected node signals are removed from the network, while keeping the remaining node signals invariant. The topology and link dynamics, or modules, of an abstracted network will generally be changed compared to the original network. Abstractions of dynamic networks can be used to select an appropriate set of node signals that are to be measured, on the basis of which a particular local module can be estimated. A method is introduced for network abstraction that generalizes previously introduced algorithms, as e.g. immersion and the method of indirect inputs. For this abstraction method it is shown under which conditions on the selected signals a particular module will remain invariant. This leads to sets of conditions on selected measured node variables that allow identification of the target module.

© 2020 Elsevier Ltd. All rights reserved.

## 1. Introduction

In current and future engineering systems, as well as in many biological and biomedical systems, large scale interconnected dynamical systems become a prime modeling target for analysis and control. The resulting dynamic networks that represent these interconnected systems, are studied from many different perspectives. In this paper our main motivation is directed towards methods and tools for data-driven modeling of (parts of) a dynamic network, extending identification methods to be able to deal with linear dynamic networks.

In network identification literature, typically three main objectives can be distinguished. One objective is to perform identification of both the topology and dynamics, see for example Sanandaji, Vincent, and Wakin (2011), Chiuso and Pillonetto (2012), Materassi and Salapaka (2012) and Zorzi and Chiuso (2017). In situations where there is prior knowledge of the topology, the objective can be to identify all dynamic modules, as

is done for example in Gonçalves and Warnick (2008), Yuan, Stan, Warnick, and Gonçalves (2011) and Weerts, Van den Hof, and Dankers (2018b), including addressing the aspect of network identifiability, see Weerts, Van den Hof, and Dankers (2018a) and Hendrickx, Gevers, and Bazanella (2019). A third possible objective is the identification of the dynamics of a single local element, or module, in the network, when the topology of the network is known. This has been addressed in Van den Hof, Dankers, Heuberger, and Bombois (2013), by extending classical closed-loop prediction error methods, known as the direct method, two-stage method and joint-io method, to the situation of dynamic networks. The reasoning has been extended in Dankers, Van den Hof, Bombois, and Heuberger (2015) to deal with signal measurements that are affected by sensor noise. Additional extensions concern identifiability aspects of local modules (Bazanella, Gevers, Hendrickx, & Parraga, 2017; Gevers, Bazanella, & Vian da Silva, 2018; Hendrickx et al., 2019; Weerts, Van den Hof, & Dankers, 2018c), and combination of the prediction error method with Bayesian estimation in Everitt, Bottegal, and Hjalmarsson (2018) and Ramaswamy, Bottegal, and Van den Hof (2018).

In this paper we will elaborate on this local module identification problem, and focus on the selection of signals to be measured in the network, to allow a particular module to be identified. While in general it may be attractive to have all node signals in a network available from measurements, in practice it may be costly or even impossible to measure some of the nodes. Then, a relevant question is which set of measured node signals is sufficient for single module identification. This problem has been addressed in several different ways.

<sup>☆</sup> This project has received funding from the European Research Council (ERC), Advanced Research Grant SYSDYNET, under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 694504), and the Vinnova Industry Excellence Center LINK-SIC, Sweden project number 2007-02224. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Alessandro Chiuso under the direction of Editor Torsten Söderström.

\* Corresponding author.

E-mail addresses: [harmweerts@gmail.com](mailto:harmweerts@gmail.com) (H.H.M. Weerts), [jonas.x.linder@se.abb.com](mailto:jonas.x.linder@se.abb.com) (J. Linder), [martin.enqvist@liu.se](mailto:martin.enqvist@liu.se) (M. Enqvist), [p.m.j.vandenhof@tue.nl](mailto:p.m.j.vandenhof@tue.nl) (P.M.J. Van den Hof).

- In [Dankers, Van den Hof, Heuberger, and Bombois \(2016\)](#) the module of interest is identified in a multi-input-single-output setting with either a direct method or a two-stage method. Signals are selected by removing non-selected signals from the network through an elimination procedure called *immersion*, while requiring that the target module remains invariant. This allows a consistent identification of the target module dynamics under appropriate excitation and disturbance conditions. A systematic way of selecting the measured node signals is provided. The selection method is extended in [Dankers, Van den Hof, Materassi, and Weerts \(2017\)](#) to handle situations of confounding variables being caused by disturbances that are spatially correlated over the different node signals. This is particularly of interest when applying the direct method, aiming at maximum likelihood results.
- The approach in [Linder \(2017\)](#) and [Linder and Enqvist \(2017a, 2017b\)](#) is also in a multi-input-single-output setting, but with an instrumental variable identification method. It follows the same philosophy as the one in [Dankers et al. \(2016\)](#), but applies a different elimination procedure, referred to as the *indirect inputs method*, and thus results in a different set of conditions on selected measured node signals.
- In [Materassi and Salapaka \(2015, 2019\)](#), selection of node signals is being done on the basis of graphical models, while Wiener filters are being used for dynamic model reconstruction, and no reference excitations are present. This approach can be characterized as an indirect approach where several dynamic objects are identified from data, after which the target module is reconstructed.
- The approach in [Bazanella et al. \(2017\)](#) and [Hendrickx et al. \(2019\)](#) is also based on indirect identification for the situation that all nodes are excited by external excitations. Here an identifiability analysis is made in order to verify whether the module of interest can be recovered uniquely from a set of estimated transfer functions from reference signals to a set of selected node signals.

Since all algorithms provide only sufficient conditions for arriving at a set of to-be-measured node signals, it is attractive to obtain results that fully characterize the degrees of freedom that are available in selecting node signals that allow appropriate module estimates.

In this paper we are going to adopt the strategy of the first two approaches, eliminating node variables to arrive at a so-called *abstracted network*, while keeping the dynamics of the target module invariant. After abstraction, the target module can be estimated on the basis of the node variables that are retained in the abstraction. While both [Dankers et al. \(2016\)](#) and [Linder and Enqvist \(2017a\)](#) have employed a particular way of abstracting networks, e.g. through network immersion (Kron reduction) or through the so-called indirect inputs method, here we develop a more general notion of abstraction that generalizes the two earlier approaches, and provides a higher flexibility in the selection mechanism of choosing which nodes to measure/retain and which nodes to discard/abstract in a multi-input single-output identification setup for estimating the target module.

In order to develop this generalized abstraction algorithm, we use the fundamental property that the network representation is not a unique representation of the behavior of the network. When manipulating the network equations, different representations can be obtained leading to different identification setups for estimating the target module. This freedom will be exploited for developing generalized conditions for selecting the node signals that are used as inputs in an identification setup for estimating the target module.

The paper is organized as follows. After defining the basis network setup in Section 2, in Section 3 the non-uniqueness of

the network representations is characterized. This is exploited in Section 4 to arrive at a generalized abstraction algorithm, which is illustrated by examples in Section 5. In Section 6 it is specified under which conditions the target module remains invariant, and consequences for the identification setup and the selection of node signals are presented in Section 7. The proofs of all results are collected in the [Appendix](#).

## 2. Dynamic network definition

In this section a dynamic network model is formulated on the basis of the setup in [Van den Hof et al. \(2013\)](#). A dynamic network consists of  $L$  scalar *internal variables* or *nodes*  $w_j, j = 1, \dots, L$ , and  $K$  *external variables*  $r_k, k = 1, \dots, K$ . Each node is a basic building block of the network and is described as:

$$w_j(t) = \sum_{\substack{l=1 \\ l \neq j}}^L G_{jl}(q)w_l(t) + u_j(t) + v_j(t) \quad (1)$$

where  $q^{-1}$  is the delay operator, i.e.  $q^{-1}w_j(t) = w_j(t-1)$ ;

- $G_{jl}$  are proper rational transfer functions that are referred to as *modules* in the network;
- There are no self-loops in the network, i.e. nodes are not directly connected to themselves  $G_{jj} = 0$ ;<sup>1</sup>
- $u_j(t)$  are generated by the *external variables*  $r_k(t)$  via

$$u_j(t) = \sum_{k=1}^K R_{jk}(q)r_k(t), \quad (2)$$

where  $r_k$  can directly be manipulated by the user, and  $R_{jk}$  are proper rational transfer functions;

- $v_j$  is *process noise*, where the vector process  $v = [v_1 \dots v_L]^T$  is modeled as a stationary stochastic process with rational spectral density, such that there exists an  $L$ -dimensional white noise process  $e := [e_1 \dots e_L]^T$ , with covariance matrix  $\Lambda > 0$  such that

$$v(t) = H(q)e(t),$$

with  $H(q)$  a proper rational transfer function.

By combining the  $L$  node signals, the network expression

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_L \end{bmatrix} = \begin{bmatrix} 0 & G_{12} & \cdots & G_{1L} \\ G_{21} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & G_{L-1,L} \\ G_{L1} & \cdots & G_{L,L-1} & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_L \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_L \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_L \end{bmatrix} \quad (3)$$

is obtained, where the zeros are due to absence of self-loops. In matrix notation the dynamic network is represented as

$$w(t) = G(q)w(t) + R(q)r(t) + H(q)e(t). \quad (4)$$

A set notation is introduced for notational convenience. Let the sets  $\mathcal{S}$  and  $\mathcal{Z}$  each contain a number of node indices, then  $w_{\mathcal{S}}$  denotes the vector of node signals consisting of all  $w_j, j \in \mathcal{S}$ . Similarly,  $G_{\mathcal{S}\mathcal{Z}}$  is the matrix of transfer functions that contains all modules  $G_{ji}, j \in \mathcal{S}, i \in \mathcal{Z}$ .

The transfer function that maps the external signals  $r$  and  $e$  into the node signals  $w$  is denoted by:

$$T(q) = \begin{bmatrix} T_{wr}(q) & T_{we}(q) \end{bmatrix}, \quad (5)$$

<sup>1</sup> Since  $G_{jj}$  are rational transfer functions, this does not limit the dynamical description of  $w_j(t)$ .

with

$$T_{wr}(q) := (I - G(q))^{-1} R(q), \text{ and} \quad (6)$$

$$T_{we}(q) := (I - G(q))^{-1} H(q). \quad (7)$$

This is also known as the open-loop response of the network corresponding with

$$w(t) = T_{wr}r(t) + \bar{v}(t), \quad (8)$$

where noise component  $\bar{v}(t)$  is defined by

$$\bar{v}(t) := T_{we}(q)e(t), \quad (9)$$

with power spectral density

$$\Phi_{\bar{v}}(\omega) := T_{we}(e^{i\omega})\Lambda T_{we}^T(e^{-i\omega}). \quad (10)$$

Some notions from graph theory will be used in the dynamic network. Modules form the interconnections/links between nodes. A node  $w_k$  is said to be an *in-neighbor* of node  $w_j$  if  $G_{jk} \neq 0$ , and then  $w_j$  is said to be an *out-neighbor* of node  $w_k$ . A *path* in a network is a sequence of interconnected nodes, more precisely there exists a path through nodes  $w_{n_1}, \dots, w_{n_k}$  if

$$G_{n_1 n_2} G_{n_2 n_3} \cdots G_{n_{k-1} n_k} \neq 0.$$

A *loop* is a path where  $n_1 = n_k$ .

A dynamic network model is then formally defined in the following way.

**Definition 1** (*Dynamic Network Model*). A network model of a network with  $L$  nodes, and  $K$  external excitation signals, with a noise process of rank  $L$  is defined by the quadruple:

$$M = (G, R, H, \Lambda)$$

with

- $G \in \mathbb{R}^{L \times L}(z)$ , diagonal entries 0;
- $R \in \mathbb{R}^{L \times K}(z)$ ;
- $H \in \mathbb{R}^{L \times L}(z)$ , monic, proper, stable, with a stable inverse;
- $\Lambda \in \mathbb{R}^{L \times L}$ ,  $\Lambda > 0$ ;
- The network is well-posed (Dankers, 2014), implying that  $(I - G)^{-1}$  exists. Additionally we require that  $(I - G)^{-1}R$  is proper and stable.  $\square$

**Remark 2.** In Definition 1 we do not require modules to be proper, while in (1) we do require properness. Real systems in practice are reflected by (1), and these will typically have proper modules. In this paper we will use the freedom to allow network models to have non-proper modules. For this reason we define a representation of these networks that allows modules to be non-proper in Definition 1. Note that for a network model representation as in Definition 1, we require  $T_{wr}$  to be proper and stable, while the noise transfer  $T_{we}$  is allowed to be non-proper, representing a non-causal mapping. This allows us to maintain a monic, proper, stable and stably invertible noise filter  $H$ , which is attractive from an identification perspective.

**Remark 3.** With the analysis provided in Weerts et al. (2018b), it is possible to extend the results in the current paper to the situation of a noise process having rank  $p$  smaller than  $L$ , implying that  $H \in \mathbb{R}^{L \times p}(z)$ , and  $\Lambda \in \mathbb{R}^{p \times p}$ .

### 3. Equivalent network representations

The freedom that is present in dynamic network representations allows for different selections of node signals to be used for identification of a module. This freedom is formally characterized in this section. Moreover, the general concept of removing a node from a network is defined as abstraction, such that in later sections we can consider abstractions that are relevant for identification.

#### 3.1. Transformation of the global network

Fundamentally, we need to define when two networks are equivalent descriptions of behavior, and what freedom is available to transform the network to an equivalent representation. In the network model definition, it has been stated that the external variables  $r$  and nodes  $w$  are known, and it is reasonable to state that equivalent networks must describe the same relation between  $r$  and  $w$ . The dynamic influence of  $r$  on  $w$  is described by the open-loop transfer function matrix  $T_{wr}$ , and so the equivalence of two networks additionally requires equality of the two related open-loop transfer function matrices from  $r$  to  $w$ . The open-loop response of the network is described by (8), i.e.  $w(t) = T_{wr}r(t) + \bar{v}(t)$ . If  $w$ ,  $r$  and  $T_{wr}$  are the same for two networks, then also  $\bar{v}$  must be the same.

**Definition 4.** Let the network model  $M^{(i)}$  correspond to open-loop transfer  $T_{wr}^{(i)}$  and noise spectrum  $\Phi_{\bar{v}}^{(i)}$  for  $i = \{1, 2\}$ . Network models  $M^{(1)}$  and  $M^{(2)}$  are said to be equivalent if

$$T_{wr}^{(1)} = T_{wr}^{(2)} \quad \text{and} \quad \Phi_{\bar{v}}^{(1)} = \Phi_{\bar{v}}^{(2)}. \quad \square \quad (11)$$

In the above definition,  $T_{wr}^{(i)}$  and  $\Phi_{\bar{v}}^{(i)}$  are associated with  $w$  and  $r$  for  $i = \{1, 2\}$ . There is an implicit assumption in the definition that the  $w$  and  $r$  are the same for both  $i = \{1, 2\}$ .

The full freedom that is available for transformation of a network model to an equivalent network model is characterized by operations applied to the network equation. These transformations can be represented by pre-multiplying with a matrix. Consider a square rational transfer function matrix  $P$ , then pre-multiplication of network equation (4) results in

$$P(q)w(t) = P(q)\left(G(q)w(t) + u(t) + v(t)\right). \quad (12)$$

The above pre-multiplication typically leads to a left-hand side unequal to  $w(t)$ , in which case we need to move terms to the right-hand side until we have  $w(t)$  on the left-hand side, i.e.

$$w(t) = (I - P(q))w(t) + P(q)\left(G(q)w(t) + u(t) + v(t)\right), \quad (13)$$

which is denoted as

$$w(t) = G^{(2)}(q)w(t) + u^{(2)}(t) + v^{(2)}(t) \quad (14)$$

where

$$G^{(2)} = I - P(I - G), \quad u^{(2)} = Pu, \quad v^{(2)} = Pv. \quad (15)$$

The transfer function matrix  $R$  is then transformed as

$$R^{(2)} = PR. \quad (16)$$

A transformation of the noise model is defined in the following way. When we describe the noise model as  $v = He$ , a pre-multiplication with  $P$  does not necessarily lead to a proper, monic, stable and stably invertible filter  $PH$ . For that reason  $H^{(2)}$  and  $\Lambda^{(2)}$  are obtained through spectral factorization of the transformed noise spectrum

$$P(e^{i\omega})\Phi_v(\omega)P^T(e^{-i\omega}) = H^{(2)}(e^{i\omega})\Lambda^{(2)}(H^{(2)}(e^{-i\omega}))^T. \quad (17)$$

A transformation  $P$  that leads to an appropriate network representation must satisfy some conditions.

**Proposition 5.** Let network model  $M^{(1)}$  satisfy Definition 1. The transformation  $P$  operating on  $M^{(1)}$  as defined in (15), (16), and (17) leads to a network model  $M^{(2)}$  that satisfies Definition 1 and that is equivalent to  $M^{(1)}$  if and only if:

- (1)  $P$  is full rank, and
- (2)  $\text{diag}(I - P(I - G^{(1)})) = 0$ .

**Proof.** Collected in the [Appendix](#).  $\square$

An interesting feature of the network transformations is that the response from external variables and process noises to internal variables remains the same. A pre-multiplication  $P$  as defined above leaves the transfer function matrix  $T_{wr}$  invariant, since

$$T_{wr} = (P(I - G))^{-1}PR, \quad (18)$$

where the identity  $P^{-1}P = I$  is used.

A pre-multiplication  $P$  that leads to a non-hollow  $G^{(2)}$  can be used to transform a network. However, in that situation additional manipulations would be necessary to arrive at a hollow representation. Without loss of generality we will restrict to transformations  $P$  that immediately result in a network representation that satisfies [Definition 1](#). There are some restrictions on  $P$ , but a large freedom in the choice of transformation  $P$  is left.

**Proposition 6.** *The equivalence transformation presented in (15), (16), and (17) can transform a network model  $M^{(1)}$  with corresponding modules  $G^{(1)}$  into a network model  $M^{(2)}$  with corresponding modules  $G^{(2)}$  using the transformation*

$$P = (I - G^{(2)})(I - G^{(1)})^{-1}. \quad (19)$$

**Proof.** Collected in the [Appendix](#).  $\square$

The proposition allows for  $G^{(1)}$  to be transformed into an arbitrary  $G^{(2)}$  as long as it is part of a valid network description. The consequence of transforming  $G^{(1)}$  to an arbitrary  $G^{(2)}$  is that the corresponding  $R^{(2)}$  will have a complex structure

$$R^{(2)} = PR^{(1)} = (I - G^{(2)})(I - G^{(1)})^{-1}R^{(1)}. \quad (20)$$

The implication is that when  $G^{(1)}$  is transformed,  $R^{(2)}$  will compensate the changes to keep the node behavior invariant. This also holds for the noise model, which will contain additional correlations. Without any further restrictions on the choice of  $R$  and  $H$ , the modules represented in  $G$  contain no information on the dynamic network. It is the combination of  $G, R, H$  that determines the dynamic network.

### 3.2. Abstraction

The next step is to extend network equivalence with the option to remove nodes from the representation. To this end the concept of *network abstraction* is defined next. This definition is related to the notions of abstraction in [Pappas and Sastry \(1997\)](#) and [Woodbury, Dankers, and Warnick \(2017\)](#).

**Definition 7.** Let network model  $M^{(1)}$  be associated with nodes  $w^{(1)} \in \mathbb{R}^{L_1}$ , external variables  $r \in \mathbb{R}^K$ , open-loop transfer  $T_{wr}^{(1)} \in \mathbb{R}^{L_1 \times K}$ , and noise spectrum  $\Phi_v^{(1)} \in \mathbb{R}^{L_1 \times L_1}$ . Let network model  $M^{(2)}$  be associated with nodes  $w^{(2)} \in \mathbb{R}^{L_2}$ , external variables  $r \in \mathbb{R}^K$ , open-loop transfer  $T_{wr}^{(2)} \in \mathbb{R}^{L_2 \times K}$ , and noise spectrum  $\Phi_v^{(2)} \in \mathbb{R}^{L_2 \times L_2}$ . Let  $L_2 < L_1$  and let  $C$  be the matrix that selects  $w^{(2)}$  from  $w^{(1)}$ , so define  $C$  with one 1 per row, zeros everywhere else, full row rank, and such that  $w^{(2)} = Cw^{(1)}$ . Network model  $M^{(2)}$  is said to be an abstraction of  $M^{(1)}$  if

$$T_{wr}^{(2)} = CT_{wr}^{(1)}, \quad \Phi_v^{(2)} = C\Phi_v^{(1)}C^T. \quad (21)$$

The nodes that are in  $w^{(1)}$ , but not in  $w^{(2)}$  are said to be abstracted from the network.  $\square$

Constructing an abstraction of a network implies that some nodes are removed from the network representation, while the remaining nodes stay invariant, in the sense that for the same external signal  $r$ , the second order statistical properties of the remaining node signals are invariant.

The next step is to determine how to obtain an abstraction of a network. In certain cases, abstracting nodes  $\bar{w}$  from a network can be done by simply pre-multiplying the network equation (4) with the selection matrix  $C$ , i.e.

$$Cw(t) = C(G(q)w(t) + R(q)r(t) + v(t)). \quad (22)$$

However, this only is an abstraction if the abstracted nodes  $\bar{w}$  no longer appear on the right-hand side of the equation. If  $\bar{w}$  appears on the right-hand side of (22) then the abstracted nodes have an influence on the behavior of the nodes in  $Cw$ , such that (21) cannot hold. It has to be determined how to define a transformation  $P$  such that an abstraction can be obtained by selecting rows from the equation, as in (22).

A node  $w_i$  influences other nodes through its out-neighbors, and these corresponding modules are located in a column in  $G$ . If a node has no influence on the rest of the network, then it has no out-neighbors, and the corresponding column is 0. Abstracting node  $w_i$  requires us to transform the network such that a 0-column is formed by transformation, after which the node can be removed. By [Proposition 6](#) we know that such a transformation always exists. The abstraction satisfies the relations

$$G^{(2)} = C(I - P(I - G))C^T, \quad R^{(2)} = CPR. \quad (23)$$

A noise model constructed as  $CPH^{(1)}$  is a non-square matrix, which is difficult to handle in an identification setting. Therefore the transformed noise model  $H^{(2)}, \Lambda^{(2)}$  will be obtained through spectral factorization

$$CP(e^{i\omega})\Phi_v(\omega)P^T(e^{-i\omega})C^T = H^{(2)}(e^{i\omega})\Lambda^{(2)}(H^{(2)}(e^{-i\omega}))^T. \quad (24)$$

### 3.3. Discussion on identifiability

In [Proposition 6](#) we have seen that  $G^{(1)}$  can be transformed into an arbitrary  $G^{(2)}$  as long as it is part of a valid network description. This may give rise to the question whether we are not dealing with an unnecessarily overparameterized situation, involving  $G, R$  and  $H$  to describe a dynamic network. However we particularly include the situation that measured external excitation signals enter into physical subsystems of a network, and thus our modules in  $G$  can have an intrinsic interpretation in the physical world. The price of abstracting nodes in a network is that  $H$  and  $R$  can become complex, which may be impossible to identify. In the identifiability analysis provided in [Weerts et al. \(2018a\)](#) the following necessary condition for network identifiability can be found: For a network model set  $\{G(\theta), R(\theta), H(\theta), \theta \in \Theta\}$  to be network identifiable it is necessary that at least  $L$  entries on each row of  $[G(q, \theta) R(q, \theta) H(q, \theta)]$  are fixed and non-parameterized. If only the topology is known, then the absent links in the network reflected by zeros in the matrices  $G, R, H$  are the only known entries. In case  $H$  and  $R$  become too complex, then the number of zeros is too small for the model to be embedded in a network identifiable model set. This implies that it is impossible to find a unique estimate for the model structure on the basis of the data. The approach in the next section is to define a particular abstraction method, that leads to abstracted networks that can be embedded in network identifiable model sets.

## 4. Abstraction of networks

We will now formulate and analyze an abstraction algorithm for dynamic networks that generalizes the procedure of immersion ([Dankers et al., 2016](#)) and the indirect inputs method ([Linder & Enqvist, 2017b](#)). It starts by dividing the network nodes into a set of nodes  $w_S$  that are retained after abstraction and a set of nodes  $w_Z$  that will be removed. The abstracted network will then allow us to analyze the properties of estimated models when the retained/measured node signals are employed in an identification procedure.

#### 4.1. Generalized algorithm

It appears that the action of abstracting nodes in a network is not unique. This is particularly due to the degrees of freedom that exist in transforming network representations to equivalent forms, by premultiplying the system's equations by appropriate transformation matrices. In order to incorporate this freedom in the abstraction, we decompose each set of node signals  $\mathcal{S}$  and  $\mathcal{Z}$  into two disjunct parts:

$$\begin{aligned} \mathcal{S} &= \mathcal{L} \cup \tilde{\mathcal{S}} \\ \mathcal{Z} &= \mathcal{V} \cup \tilde{\mathcal{Z}}. \end{aligned} \quad (25)$$

The node signals  $w_{\tilde{\mathcal{Z}}}$  will be abstracted directly by substituting the equation for  $w_{\tilde{\mathcal{Z}}}$  into the equations for the other node signals. The node signals  $w_{\mathcal{V}}$  have the property that they can be *indirectly observed* by the node signals  $w_{\mathcal{L}}$ , and therefore they can be eliminated from the network by utilizing the equation for nodes  $w_{\mathcal{L}}$ . The notion of *indirect observation* will be specified after the next step.

Based on these sets, the network can be represented by

$$\begin{bmatrix} w_{\tilde{\mathcal{S}}} \\ w_{\mathcal{L}} \\ w_{\mathcal{V}} \\ w_{\tilde{\mathcal{Z}}} \end{bmatrix} = \begin{bmatrix} G_{\tilde{\mathcal{S}}\tilde{\mathcal{S}}} & G_{\tilde{\mathcal{S}}\mathcal{L}} & G_{\tilde{\mathcal{S}}\mathcal{V}} & G_{\tilde{\mathcal{S}}\tilde{\mathcal{Z}}} \\ G_{\mathcal{L}\tilde{\mathcal{S}}} & G_{\mathcal{L}\mathcal{L}} & G_{\mathcal{L}\mathcal{V}} & G_{\mathcal{L}\tilde{\mathcal{Z}}} \\ G_{\mathcal{V}\tilde{\mathcal{S}}} & G_{\mathcal{V}\mathcal{L}} & G_{\mathcal{V}\mathcal{V}} & G_{\mathcal{V}\tilde{\mathcal{Z}}} \\ G_{\tilde{\mathcal{Z}}\tilde{\mathcal{S}}} & G_{\tilde{\mathcal{Z}}\mathcal{L}} & G_{\tilde{\mathcal{Z}}\mathcal{V}} & G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}} \end{bmatrix} \begin{bmatrix} w_{\tilde{\mathcal{S}}} \\ w_{\mathcal{L}} \\ w_{\mathcal{V}} \\ w_{\tilde{\mathcal{Z}}} \end{bmatrix} + \begin{bmatrix} u_{\tilde{\mathcal{S}}} \\ u_{\mathcal{L}} \\ u_{\mathcal{V}} \\ u_{\tilde{\mathcal{Z}}} \end{bmatrix} + \begin{bmatrix} v_{\tilde{\mathcal{S}}} \\ v_{\mathcal{L}} \\ v_{\mathcal{V}} \\ v_{\tilde{\mathcal{Z}}} \end{bmatrix}, \quad (27)$$

where  $w_{\tilde{\mathcal{Z}}}$  and  $w_{\tilde{\mathcal{S}}}$  are defined according to  $\tilde{\mathcal{S}} = \mathcal{S} \setminus \mathcal{L}$  and  $\tilde{\mathcal{Z}} = \mathcal{Z} \setminus \mathcal{V}$ .

The node signals  $w_{\mathcal{L}}$  and  $w_{\mathcal{V}}$  are chosen in such a way that the signals  $w_{\mathcal{L}}$  serve as indirect observations of the node signals  $w_{\mathcal{V}}$ , meaning that the signals  $w_{\mathcal{L}}$  contain sufficient information of the indirectly observed signals  $w_{\mathcal{V}}$ , so as to replace them in an elimination procedure. This is formulated in the following definition.

**Definition 8 (Indirect Observations).** The node signals  $w_{\mathcal{L}}$  serve as indirect observations of the node signals  $w_{\mathcal{V}}$  if the transfer function  $G_{\mathcal{L}\mathcal{V}} + G_{\mathcal{L}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1}G_{\tilde{\mathcal{Z}}\mathcal{V}}$  has full column rank.  $\square$

The property is satisfied if in the network there exists a sufficient number of paths from nodes  $w_{\mathcal{V}}$  to nodes  $w_{\mathcal{L}}$  that run through nonmeasured nodes only. An illustration of indirect observations is provided in Section 4.3. In the remainder of this paper it will be assumed that  $w_{\mathcal{L}}$  and  $w_{\mathcal{V}}$  are selected to satisfy the full column rank property of the above definition. Note that this assumption is not restrictive because  $\mathcal{L}$  and  $\mathcal{V}$  are chosen by the user such that the assumption is satisfied. We can always choose  $\mathcal{L}$  and  $\mathcal{V}$  as empty sets, so the assumption does not prevent us from constructing an abstracted network. However, the use of non-empty sets  $\mathcal{L}$  and  $\mathcal{V}$  will allow us to use more degrees of freedom in constructing abstracted networks.

Now we can formulate the generalized abstraction algorithm, as follows.

**Algorithm 1.** Consider a network representation as in (27). Then the following algorithm leads to a network in which nodes  $w_{\mathcal{Z}}$  are abstracted and nodes  $w_{\mathcal{S}}$  are retained:

- Solve the fourth equation of (27) for  $w_{\tilde{\mathcal{Z}}}$ , and then substitute the result into the other equations, and remove the fourth equation from the network.
- Solve the second equation of (27) for  $w_{\mathcal{V}}$ , and substitute the result into the first equation.
- Solve the third equation of (27) for  $w_{\mathcal{V}}$ , and substitute the result into the second equation, and remove the third equation from the network.

- Remove possible self-loops in the resulting network representation by shifting self-loop terms to the left hand side of the equations and scaling the equations such that an identity matrix remains at the left hand side.  $\square$

The algorithm shows that the essential difference between the nodes in  $w_{\tilde{\mathcal{S}}}$  and  $w_{\mathcal{L}}$  is how  $w_{\mathcal{V}}$  is removed from their equation. After application of the algorithm, the abstracted network will be represented by

$$\begin{bmatrix} w_{\tilde{\mathcal{S}}} \\ w_{\mathcal{L}} \end{bmatrix} = \begin{bmatrix} \check{G}_{\tilde{\mathcal{S}}\tilde{\mathcal{S}}} & \check{G}_{\tilde{\mathcal{S}}\mathcal{L}} \\ \check{G}_{\mathcal{L}\tilde{\mathcal{S}}} & \check{G}_{\mathcal{L}\mathcal{L}} \end{bmatrix} \begin{bmatrix} w_{\tilde{\mathcal{S}}} \\ w_{\mathcal{L}} \end{bmatrix} + \begin{bmatrix} \check{u}_{\tilde{\mathcal{S}}} \\ \check{u}_{\mathcal{L}} \end{bmatrix} + \begin{bmatrix} \check{v}_{\tilde{\mathcal{S}}} \\ \check{v}_{\mathcal{L}} \end{bmatrix}. \quad (28)$$

Note that the particular type and ordering of variable substitution in Algorithm 1 essentially influences the result. This will be illustrated with examples in Section 5.

#### 4.2. Specification through transformations

Algorithm 1 can be specified by denoting the algebraic manipulations that generate the substitution and elimination operations in the different steps of the algorithm. We will first describe the substitution operations on the set of equations, and at the end of the procedure address the removal of equations.

In step (a) the elimination of  $w_{\tilde{\mathcal{Z}}}$  is performed, which corresponds to applying the transformation matrix

$$P^{(1)} = \begin{bmatrix} I & 0 & 0 & G_{\tilde{\mathcal{S}}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1} \\ 0 & I & 0 & G_{\mathcal{L}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1} \\ 0 & 0 & I & G_{\mathcal{V}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1} \\ 0 & 0 & 0 & (I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1} \end{bmatrix}, \quad (29)$$

to the network representation (27). This leads to a new  $G$ -matrix given by

$$G^{(1)} = I - P^{(1)}(I - G), \quad (30)$$

where  $G$  is partitioned as defined in (27), and  $G^{(1)}$  is partitioned in the same way.

In steps (b)–(c), we first obtain a new expression for  $w_{\mathcal{V}}$  by reverting the expression for  $w_{\mathcal{L}}$ , and we substitute the original expression for  $w_{\mathcal{V}}$  into the expressions for  $w_{\mathcal{L}}$ . This corresponds to applying the transformation matrix

$$P^{(2)} = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & G_{\mathcal{L}\mathcal{V}}(I - G_{\mathcal{V}\mathcal{V}}^{(1)})^{-1} & 0 \\ 0 & (G_{\mathcal{L}\mathcal{V}}^{(1)})^\dagger & 0 & 0 \\ 0 & 0 & 0 & I \end{bmatrix}, \quad (31)$$

with  $(G_{\mathcal{L}\mathcal{V}}^{(1)}) = G_{\mathcal{L}\mathcal{V}} + G_{\mathcal{L}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1}G_{\tilde{\mathcal{Z}}\mathcal{V}}$ , and  $(G_{\mathcal{L}\mathcal{V}}^{(1)})^\dagger$  denoting its left-inverse, and  $(G_{\mathcal{V}\mathcal{V}}^{(1)}) = G_{\mathcal{V}\mathcal{V}} + G_{\mathcal{V}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1}G_{\tilde{\mathcal{Z}}\mathcal{V}}$ , leading to

$$G^{(2)} = I - P^{(2)}(I - G^{(1)}), \quad (32)$$

where  $G^{(2)}$  has the same partitioning as  $G^{(1)}$ . Note that the left-inverse exists due to the indirect observations property of  $w_{\mathcal{L}}$ , as formulated in Definition 8.

The remaining part of steps (b)–(c) is now to substitute the new expression for  $w_{\mathcal{V}}$  in the first equation for  $w_{\tilde{\mathcal{S}}}$ , thereby eliminating the dependency of this expression on  $w_{\mathcal{V}}$ . This is achieved by applying the transformation matrix

$$P^{(3)} = \begin{bmatrix} I & 0 & G_{\tilde{\mathcal{S}}\mathcal{V}}^{(2)} & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & G_{\tilde{\mathcal{Z}}\mathcal{V}}^{(2)} & I \end{bmatrix}, \quad (33)$$

such that

$$G^{(3)} = I - P^{(3)}(I - G^{(2)}), \quad (34)$$

where  $G^{(3)}$  has the same partitioning as  $G^{(2)}$ . The additional term  $G_{\tilde{z}\nu}^{(2)}$  that is added in the fourth row of  $P^{(3)}$  ensures that in the transformed network all columns that correspond to  $w_{\tilde{z}}$  are zero in  $G^{(4)}$ , including for the equations that will be removed.

Step (d) of the Algorithm is addressed by removing self-loops in the resulting network representation, by applying a diagonal transformation matrix  $P^{(4)}$  with diagonal elements

$$P_{jj}^{(4)} = \frac{1}{1 - G_{jj}^{(3)}} \quad (35)$$

and being 0 elsewhere.

The total transformation that is applied to the network representation is now given by

$$P^{(abs)} = P^{(4)}P^{(3)}P^{(2)}P^{(1)}, \quad (36)$$

which leads to a  $G$ -matrix of the transformed network representation that is structured according to

$$G^{(4)} := (I - P^{(abs)}(I - G)) = \begin{bmatrix} \check{G}_{\tilde{s}\tilde{s}} & \check{G}_{\tilde{s}\tilde{z}} & 0 & 0 \\ \check{G}_{\tilde{z}\tilde{s}} & \check{G}_{\tilde{z}\tilde{z}} & 0 & 0 \\ \check{G}_{\nu\tilde{s}} & \check{G}_{\nu\tilde{z}} & 0 & 0 \\ \check{G}_{\tilde{z}\nu} & \check{G}_{\tilde{z}\nu} & 0 & 0 \end{bmatrix}. \quad (37)$$

The abstracted network now results by selecting the first two block rows and columns in the matrix  $G^{(4)}$ , thereby removing the equations for the unmeasured/abstracted node variables  $w_{\nu}$  and  $w_{\tilde{z}}$ .

**Proposition 9.** *When applying the abstraction procedure of Algorithm 1 to a dynamic network given by (27), the obtained abstracted network is the same as the abstracted network given by (28) with*

$$\begin{bmatrix} \check{G}_{\tilde{s}\tilde{s}} & \check{G}_{\tilde{s}\tilde{z}} \\ \check{G}_{\tilde{z}\tilde{s}} & \check{G}_{\tilde{z}\tilde{z}} \end{bmatrix} = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \end{bmatrix} (I - P^{(abs)}(I - G)) \begin{bmatrix} I & 0 \\ 0 & I \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

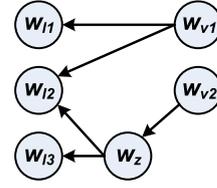
$$\begin{bmatrix} \check{u}_{\tilde{s}} \\ \check{u}_{\tilde{z}} \end{bmatrix} = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \end{bmatrix} P^{(abs)} \begin{bmatrix} u_{\tilde{s}} \\ u_{\tilde{z}} \\ u_{\nu} \\ u_{\tilde{z}} \end{bmatrix}$$

$$\begin{bmatrix} \check{v}_{\tilde{s}} \\ \check{v}_{\tilde{z}} \end{bmatrix} = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \end{bmatrix} P^{(abs)} \begin{bmatrix} v_{\tilde{s}} \\ v_{\tilde{z}} \\ v_{\nu} \\ v_{\tilde{z}} \end{bmatrix}.$$

**Proof.** Collected in the Appendix.  $\square$

#### 4.3. Interpretations and discussion

Compared to selecting a set of nodes  $w_{\tilde{z}}$ , the particular choice of the sets of nodes  $w_{\tilde{z}}$  and  $w_{\nu}$  creates additional degrees of freedom in the problem of constructing an abstracted network, in which the nodes  $w_{\tilde{z}}$  (including  $w_{\nu}$ ) are removed. The mechanism that is used is that the network equation for the node signals  $w_{\tilde{z}}$  is inverted to become an equation that describes the node signals  $w_{\nu}$ . This equation is then subsequently used to substitute and eliminate the  $w_{\nu}$  signals from the abstracted network. In order to be able to use the network equation for  $w_{\tilde{z}}$  in this way, it needs to capture full information on the node signals  $w_{\nu}$ . This is reflected in the property of *indirect observations*, and the required full column rank property of  $G_{\tilde{z}\nu} + G_{\tilde{z}\tilde{z}}(I - G_{\tilde{z}\tilde{z}})^{-1}G_{\tilde{z}\nu}$ , as formulated in Definition 8. This full rank property implies that  $\dim(w_{\tilde{z}}) \geq \dim(w_{\nu})$ . It is generically satisfied if there are  $\dim(w_{\nu})$  vertex-disjoint paths from  $w_{\nu}$  to  $w_{\tilde{z}}$  that run through nonmeasured/abstracted nodes only (Hendrickx et al., 2019; van der Woude, 1991). An example of the full rank



**Fig. 1.** Example network with  $\nu = \{v_1, v_2\}$ ,  $\tilde{z} = \{z\}$ ,  $\mathcal{L} = \{l_1, l_2, l_3\}$ , where the full rank condition is satisfied.

assumption is shown in Fig. 1. In the figure there are the two vertex-disjoint paths  $w_{v_1} \rightarrow w_{l_1}$  and  $w_{v_2} \rightarrow w_z \rightarrow w_{l_3}$  for two nodes that are indirectly observed. In this case actually any selection of two nodes from  $\{w_{l_1}, w_{l_2}, w_{l_3}\}$  would be sufficient to act as indirect observations of  $\{w_{v_1}, w_{v_2}\}$ .

The network abstraction introduced here, generalizes two earlier introduced abstraction algorithms. For the particular choice,  $\mathcal{L} = \emptyset$  and  $\nu = \emptyset$ , Algorithm 1 describes the method of network immersion, as introduced in Dankers et al. (2016), and developed for the situation  $R = I$ . In that case steps (2)–(3) of the algorithm become obsolete.

If  $w_{\tilde{z}}$  is restricted to consist of nodes that are out-neighbors of  $w_{\nu}$ , and  $w_{\nu}$  does not contain in-neighbors of  $w_{\tilde{z}}$ , and  $G_{\tilde{z}\nu}$  has full column rank, then Algorithm 1 describes the indirect inputs method as defined in Linder and Enqvist (2017a), which has been developed for the situation  $R = I$ . If  $\mathcal{L} = \emptyset$ , then the indirect inputs method is equivalent to the immersion method.

Because of well-posedness of the original network, all terms in the transformation matrices  $P^{(1)} \dots P^{(4)}$  are proper, except possibly for the term  $(G_{\tilde{z}\nu}^{(1)})^\dagger$  which may be non-proper. This implies that the introduction of the sets  $\mathcal{L}$  and  $\nu$  may lead to a final abstracted network representation that is non-proper. Properness of the resulting network representation is guaranteed if  $\mathcal{L} = \emptyset$ .

#### 4.4. Identifiability analysis

As discussed in Section 3.3, it is impossible to formulate an identifiable model set for network representations of high complexity. The underlying objective of the particular abstraction algorithm is to limit the complexity of the abstracted network. An evaluation of the structure of network representations obtained with the abstraction algorithm is made. In this way we can guarantee that an identifiable model set can be defined for the abstracted network.

A sufficient condition for network identifiability that can easily be verified is that every node has an independent external excitation. This is achieved when the columns of  $\check{R}$  in

$$\check{u}(t) = \check{R}(q)r \quad (38)$$

can be permuted to arrive at a matrix with a leading diagonal (Weerts et al., 2018a). In order to verify whether this can be achieved we need to evaluate the structure of  $\check{R}$ . The abstracted network generated by Algorithm 1 corresponds to the transformation  $P^{(abs)}$ , such that

$$\check{R} = \begin{bmatrix} I & 0 \end{bmatrix} P^{(abs)} R. \quad (39)$$

Since  $P^{(abs)}$  and  $R$  are formulated in terms of the original network, we can formulate conditions for network identifiability based on the original topology. This is formally done in the next proposition.

**Proposition 10.** *Consider the abstracted network (28) obtained by abstracting the original network (4) with Algorithm 1 by using the sets of nodes  $\tilde{s}, \mathcal{L}, \nu, \tilde{z}$ . The representation of the external excitations is  $\check{u}(t) = \check{R}(q)r$ . The matrix  $\check{R}(q)$  can be given a leading diagonal by column operations if the original network is such that*

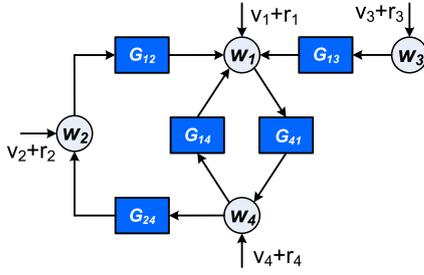


Fig. 2. Example network to illustrate abstraction.

- (1)  $R_{\tilde{S}\tilde{S}}$  is diagonal, and  $r_{\tilde{S}}$  is not an in-neighbor of nodes other than  $w_{\tilde{S}}$ ,
- (2)  $R_{\mathcal{V}\mathcal{V}}$  is diagonal, and  $r_{\mathcal{V}}$  is not an in-neighbor of nodes other than  $w_{\mathcal{V}}$ , and
- (3)  $G_{\mathcal{L}\mathcal{V}}$  is diagonal,  $G_{\mathcal{L}\tilde{Z}} = 0$ ,  $G_{\mathcal{V}\mathcal{V}} = 0$ , and  $G_{\mathcal{V}\tilde{Z}} = 0$ .

**Proof.** Collected in the [Appendix](#).  $\square$

The proposition implies that abstracted networks that are obtained by [Algorithm 1](#) can be embedded in network identifiable model sets, under some restrictions on the original network. Here we have analyzed network identifiability of all modules in the abstracted network using sufficient conditions. The result may be extended by using less restrictive conditions that make use of the topology present in  $G$  ([Weerts et al., 2018a](#)). Moreover, for consistency of the module of interest only network identifiability of that particular module is necessary. Conditions for network identifiability of a particular module are less restrictive than conditions for network identifiability of all modules ([Weerts et al., 2018c](#)), which could further reduce the imposed conditions on the structure of the network.

## 5. Abstraction applied to an example network

In this section we will provide an example to illustrate some of the options that are available in network abstraction. Consider the network in [Fig. 2](#) where the nodes are described by the following equations

$$w_1 = G_{12}w_2 + G_{13}w_3 + G_{14}w_4 + r_1 + v_1 \quad (40)$$

$$w_2 = G_{24}w_4 + r_2 + v_2 \quad (41)$$

$$w_3 = r_3 + v_3 \quad (42)$$

$$w_4 = G_{41}w_1 + r_4 + v_4. \quad (43)$$

If we would like to abstract node  $w_4$ , e.g. because this node signal cannot be measured, then we have different options for doing so. The set of retained nodes is  $\mathcal{S} = \{1, 2, 3\}$  and the set of removed nodes is  $\mathcal{Z} = \{4\}$ .

We have to make a choice on whether nodes are used as indirect observations. If we choose that there are no indirect observations, then  $\mathcal{L} = \mathcal{V} = \emptyset$ , such that  $\tilde{\mathcal{S}} = \{1, 2, 3\}$  and  $\tilde{\mathcal{Z}} = \{4\}$ . We eliminate  $w_4$  from the system equations by substituting its expression (43) in the expressions of the nodes that are retained (40)–(42). This leads to a new set of equations given by

$$w_1 = G_{14}G_{41}w_1 + G_{12}w_2 + G_{13}w_3 + r_1 + v_1 + G_{14}(r_4 + v_4)$$

$$w_2 = G_{24}G_{41}w_1 + r_2 + v_2 + G_{24}(r_4 + v_4)$$

$$w_3 = r_3 + v_3$$

which induces a self-loop around  $w_1$ . This can be compensated for by moving the  $w_1$ -dependent term to the left hand side, and rewriting the equation for  $w_1$  as

$$w_1 = S[G_{12}w_2 + G_{13}w_3 + r_1 + v_1 + G_{14}(r_4 + v_4)]$$

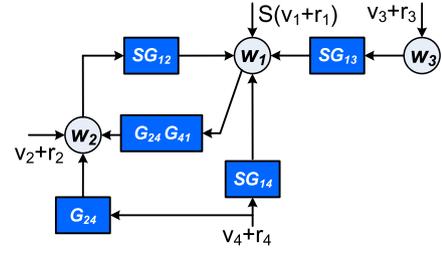


Fig. 3. Network obtained after abstracting node  $w_4$  through immersion.

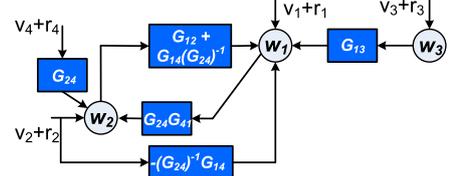


Fig. 4. Modification of the network depicted in [Fig. 2](#), obtained after removal of node  $w_4$  by the indirect inputs method.

with  $S := (1 - G_{14}G_{41})^{-1}$ . As a result the abstracted network is obtained and sketched in [Fig. 3](#). This way of eliminating node  $w_4$  is referred to as *immersion* ([Dankers et al., 2016](#)), and comes down to lifting each path in the original network that contains the node signal that is eliminated. After removing the abstracted node signals, the remaining node signals are invariant.

As an alternative for the chosen abstraction, we can choose  $w_2$  as an indirect observation of  $w_4$ , such that  $\mathcal{L} = \{2\}$ ,  $\mathcal{V} = \{4\}$ ,  $\tilde{\mathcal{S}} = \{1, 3\}$ , and  $\tilde{\mathcal{Z}} = \emptyset$ . In this situation, node signal  $w_4$  is abstracted by utilizing the expression for  $w_2$ . We rewrite [Eq. \(41\)](#) as

$$w_4 = G_{24}^{-1}(w_2 - v_2 - r_2)$$

and substitute this into (40) to obtain the expression for  $w_1$

$$w_1 = G_{12}w_2 + G_{13}w_3 + G_{14} \underbrace{G_{24}^{-1}(w_2 - v_2 - r_2)}_{w_4} + r_1 + v_1.$$

In order to obtain the new expression for  $w_2$  we directly substitute the expression for  $w_4$  (43) into the expression for  $w_2$  (41). The abstracted network is sketched in [Fig. 4](#), and given by the following equations:

$$w_1 = (G_{12} + G_{14}G_{24}^{-1})w_2 + G_{13}w_3 + r_1 + v_1$$

$$- G_{14}G_{24}^{-1}(r_2 + v_2)$$

$$w_2 = G_{24}G_{41}w_1 + r_2 + v_2 + G_{24}(r_4 + v_4)$$

$$w_3 = r_3 + v_3.$$

This alternative method of eliminating node variable  $w_4$  is referred to as the *indirect inputs method* introduced in [Linder and Enqvist \(2017b\)](#). The principle idea is that the out-neighbor of a node that needs to be abstracted contains information about that node. Then the equation of the out-neighbor is manipulated in order to obtain an explicit expression for the node to be abstracted, which is then used to eliminate the node from the network. A major difference with the method of immersion is that the inverse of modules may appear in the resulting network representation.

It can be observed that the network topology and module dynamics can change when nodes are abstracted from the network. In particular the module  $G_{13}$  has changed to  $SG_{13}$  when the immersion method is applied, while it remains invariant when the indirect inputs method is applied. This is going to be

important when considering the problem of identifying a local module on the basis of a restricted set of measured node signals, as will be discussed in the next section.

## 6. Identification setting for invariant modules

We have introduced an algorithm to perform network abstraction. The remaining question to answer is how this will help to select nodes for the identification of the module of interest. A central point in our reasoning will be the invariance of the target module in the abstracted network. Although for consistent identification of the target module it is not strictly necessary to have target module invariance, cf. e.g. the indirect type identification methods of Bazanella et al. (2017), Gevers et al. (2018) and Hendrickx et al. (2019) or the Wiener-filter based method of Materassi and Salapaka (2015, 2019), invariant target modules are very attractive in two-stage methods (Dankers et al., 2016; Linder & Enqvist, 2017b), and they are indispensable in direct methods (Dankers et al., 2016) that have the potential to provide consistent and maximum likelihood (and thus minimum variance) results. In Dankers et al. (2016) and Linder and Enqvist (2017b) two different abstraction methods have been used to select node signals to be measured for identification of a target module, based on the invariant module principle. The prime reasoning and the formulation of generalized results are presented next.

### 6.1. Direct identification setup

If the target module to be identified is  $G_{ji}(q)$ , then a MISO identification setup on the basis of the abstracted network can be formulated in the following way. Node  $w_j$  is used as output, and the following nodes are inputs:  $w_{\mathcal{S} \setminus j}$  with  $i \in \mathcal{S}$ ,  $w_{\mathcal{L}}$ , and possibly additional external excitations. In this way an identification algorithm can provide us with a consistent estimate of the modules of the abstracted network  $\check{G}_{jk}$  for all  $k \in \mathcal{S}$ , provided that some regularity conditions are satisfied, among which sufficient excitation properties of the measured signals. We have seen in the examples of the previous section that a module may remain unchanged after abstraction for particular choices of the sets  $\check{\mathcal{S}}$ ,  $\check{\mathcal{L}}$ ,  $\check{\mathcal{V}}$ ,  $\check{\mathcal{Z}}$ , i.e.

$$\check{G}_{ji}(q) = \check{G}_{ji}(q). \quad (44)$$

If the modules of the abstracted network are estimated consistently, and the module of interest has remained invariant in the abstracted network, then the module of interest is estimated consistently. In the remainder of this section we will address the problem under which conditions the abstracted network has the mentioned invariance property (44) of  $G_{ji}$ .

### 6.2. Invariance of the module $G_{ji}$

When applying immersion as a specific abstraction algorithm, it has been analyzed in Dankers et al. (2016) under which conditions on the set of retained node signals, a particular module in the network will remain invariant.

**Proposition 11** (Dankers et al., 2016). *Consider a dynamic network as defined in (4), and let  $G_{ji}(q)$  be the module of interest. Denote with  $\check{G}_{ji}(q, \check{\mathcal{S}}, \check{\mathcal{L}}, \check{\mathcal{V}}, \check{\mathcal{Z}})$  the module  $\check{G}_{ji}$  in the network that is abstracted using Algorithm 1 with the sets  $\check{\mathcal{S}}, \check{\mathcal{L}}, \check{\mathcal{V}}, \check{\mathcal{Z}}$ . Let the network abstraction be performed through immersion, i.e. with  $\mathcal{L} = \mathcal{V} = \emptyset$ . Define the set  $\mathcal{D}_j = \check{\mathcal{S}} \setminus j$ . Then*

$$\check{G}_{ji}(q, \check{\mathcal{S}}, \check{\mathcal{L}}, \check{\mathcal{V}}, \check{\mathcal{Z}}) = G_{ji}(q) \quad (45)$$

if  $\mathcal{D}_j$  satisfies the following conditions:

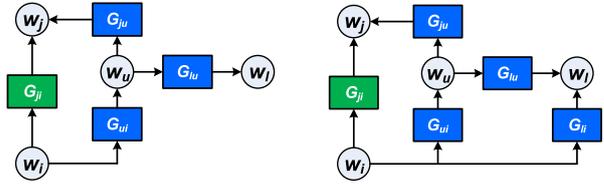


Fig. 5. Networks to illustrate issues with parallel paths when abstracting.

- (1)  $i \in \mathcal{D}_j, j \notin \mathcal{D}_j$ ,
- (2) every path from  $w_i$  to  $w_j$ , excluding the path  $G_{ji}$ , goes through a node  $w_k, k \in \mathcal{D}_j$ ,
- (3) every loop from  $w_j$  to  $w_j$  goes through a node  $w_k, k \in \mathcal{D}_j$ .  $\square$

According to this proposition, there are two situations that need to be checked for guaranteeing module invariance: parallel paths and loops around the output. Every path that connects input and output parallel to the target module, and every loop around the output should be “blocked” by another node that is retained in the abstracted network.

### 6.3. Generalization of invariance conditions

The following two examples illustrate the parallel paths and loops around the outputs, leading to a generalization of Proposition 11 that extends the applicability from the immersion abstraction algorithm to the generalized abstraction algorithm. In the next examples, noise-free networks are used in order to stick to the core reasoning.

**Example 12 (Parallel Paths).** Consider the left network in Fig. 5 and the module of interest  $G_{ji}$ . Paths that run in parallel to this module, i.e. paths from  $w_i$  to  $w_j$ , may lead to changes in the module of interest during abstraction. If  $w_u$  is removed using immersion, with  $\mathcal{L} = \emptyset, \mathcal{V} = \emptyset$ , then  $w_u = G_{ui}w_i$  is substituted into the equation for  $w_j$ , such that the dynamics of modules  $G_{ju}$  and  $G_{ui}$  are merged with module of interest  $w_i \rightarrow w_j$ , i.e.

$$w_j = (G_{ji} + G_{ju}G_{ui})w_i. \quad (46)$$

As stated in Proposition 11, a way to prevent these parallel paths from changing the module of interest is by including a node in every parallel path in the abstracted network, for example by measuring  $w_u$ .

An alternative way of removing  $w_u$  is to include  $w_l$  as an indirect observation of  $w_u$ , i.e. by choosing  $w_{\mathcal{L}} = w_l, w_{\mathcal{V}} = w_u$ . In this case the node  $w_u$  is substituted with  $w_u = G_{lu}^{-1}w_l$  such that

$$w_j = G_{ji}w_i + G_{ju}G_{lu}^{-1}w_l. \quad (47)$$

The substitution uses an equation that does not contain  $w_i$ , such that  $G_{ji}$  remains invariant. Apparently, it is not strictly necessary to include a node in every parallel path in the abstracted network. An indirect observation of a node in the parallel path may be used to block this path.

When an additional path  $w_i \rightarrow w_l$  exists as in the right network in Fig. 5, the situation changes. Now the equation for node  $w_l$  depends on  $w_i$ , and if the unknown node  $w_u$  is eliminated using the indirect observation  $w_l$ , then an additional contribution from  $w_i$  appears such that the module of interest is changed, i.e.

$$w_j = (G_{ji} - G_{ju}G_{lu}^{-1}G_{li})w_i + G_{ju}G_{lu}^{-1}w_l, \quad (48)$$

where  $w_u = G_{lu}^{-1}(w_l - G_{li}w_i)$  is used. If in the left network of Fig. 5 there is no path from  $w_u$  to  $w_l$ , then  $w_l$  cannot be used as an indirect observation.  $\square$

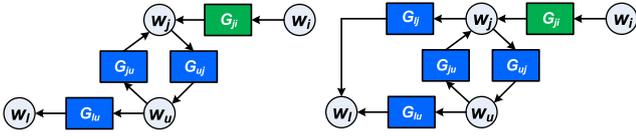


Fig. 6. Networks to indicate issues with self-loops when making abstractions.

From the example it can be observed that the nodes used as indirect observations, i.e.  $w_{\mathcal{L}}$ , should not have  $w_i$  as an in-neighbor.

**Example 13 (Self-loops).** Consider the left network in Fig. 6 and suppose the module of interest is  $G_{ji}$ . Paths that run as a loop around the output of this module, i.e. paths from  $w_j$  to  $w_j$ , may lead to changes in the module of interest during abstraction. If the node  $w_u$  of the left network in Fig. 6 is eliminated by immersion, using  $\mathcal{L} = \emptyset$  and  $\mathcal{V} = \emptyset$ , then abstraction leads to the following. The equation  $w_u = G_{uj}w_j$  is substituted into the equation for  $w_j$ , after which a self-loop around  $w_j$  is resolved. This leads to the following change in the module of interest

$$w_j = \frac{G_{ji}}{1 - G_{ju}G_{uj}}w_i. \quad (49)$$

As stated in Proposition 11, a way to prevent these loops around the output from changing the module of interest is by including a node in every such loop around  $w_j$  in the abstracted network, for example by measuring  $w_u$ .

An alternative way of removing  $w_u$  is to include  $w_l$  as an indirect observation of  $w_u$ , i.e. by choosing  $w_{\mathcal{L}} = w_l$  and  $w_{\mathcal{V}} = w_u$ . In this case the  $w_u$  is substituted for  $w_u = G_{lu}^{-1}w_l$  such that

$$w_j = G_{ji}w_i + G_{ju}G_{lu}^{-1}w_l. \quad (50)$$

The substitution uses an equation that does not contain  $w_j$ , such that no self-loop has to be resolved, and  $G_{ji}$  remains invariant. It is thus not strictly necessary to include a node in every loop around  $w_j$  in the abstracted network. An indirect observation of a node in a loop around  $w_j$  may be used to block this path.

If instead there is a direct link  $w_j \rightarrow w_l$  like in the right network of Fig. 6, then  $w_l$  depends directly on  $w_j$ , and using this equation for elimination of  $w_u$  would again lead to a dependence of  $w_j$  on itself in the abstracted network, i.e.

$$w_j = G_{ji}w_i + G_{ju}G_{lu}^{-1}(w_l - G_{lj}w_j), \quad (51)$$

where  $w_u$  is substituted for  $w_u = G_{lu}^{-1}(w_l - G_{lj}w_j)$ . The self-loop should be resolved, leading to

$$w_j = \frac{G_{ji}}{1 + G_{ju}G_{lu}^{-1}G_{lj}}w_i + \frac{G_{ju}G_{lu}^{-1}}{1 + G_{ju}G_{lu}^{-1}G_{lj}}w_l \quad (52)$$

where it is obvious that the module of interest has changed.  $\square$

In conclusion, for verifying module invariance in abstracted networks obtained by Algorithm 1 we have to consider the following. It is not sufficient to only consider parallel paths from  $w_i$  to  $w_j$  and loops from  $w_j$  to  $w_j$  that appear in the data generating system. We have to also consider indirect observations of the nodes that are part of parallel paths and loops around the output. Paths from  $w_i$  and  $w_j$  to the indirect observations  $w_{\mathcal{L}}$  also have to be considered to avoid merging of paths and to keep  $G_{ji}$  invariant under the transformation. These observations lead to the following formal result.

**Theorem 14.** Consider a dynamic network as defined in (4), and let  $G_{ji}(q)$  be the module of interest. Denote with  $\check{G}_{ji}(q, \tilde{\mathcal{S}}, \mathcal{L}, \mathcal{V}, \tilde{\mathcal{Z}})$  the module  $\check{G}_{ji}$  in the abstracted network that is obtained using

Algorithm 1 with the sets  $\tilde{\mathcal{S}}, \mathcal{L}, \mathcal{V}, \tilde{\mathcal{Z}}$ . Assume that nodes  $w_{\mathcal{L}}$  act as indirect observations of nodes  $w_{\mathcal{V}}$  according to Definition 8, and that  $\{i, j\} \subset \tilde{\mathcal{S}}$ . Define the sets  $\mathcal{T} = \{j\} \cup \mathcal{L}$  and  $\mathcal{K} = \mathcal{V} \cup \tilde{\mathcal{S}} \setminus \{j\}$ . Then

$$\check{G}_{ji}(q, \tilde{\mathcal{S}}, \mathcal{L}, \mathcal{V}, \tilde{\mathcal{Z}}) = G_{ij}(q) \quad (53)$$

if the following conditions on the sets  $\tilde{\mathcal{S}}, \mathcal{L}$  and  $\mathcal{V}$  are satisfied:

- All paths from  $w_i$  to  $w_{\mathcal{T}}$ , excluding the direct path  $G_{ji}$ , pass through a node  $w_k$ ,  $k \in \mathcal{K} \setminus \{i\}$ ,
- All paths from  $w_j$  to  $w_{\mathcal{T}}$  pass through a node  $w_k$ ,  $k \in \mathcal{K}$ .

**Proof.** Collected in the Appendix.  $\square$

In condition (a) the index  $i$  is excluded from the set  $\mathcal{K}$  since every path that starts in  $w_i$  contains a node in  $\mathcal{K}$ . Conditions (a) and (b) imply that there cannot be any direct paths from  $w_i$  and  $w_j$  to indirect observations  $w_{\mathcal{L}}$ , i.e.  $G_{\mathcal{L}i} = 0$ , and  $G_{\mathcal{L}j} = 0$ .

The set  $\mathcal{K}$  is the set of, either directly retained signals in  $\tilde{\mathcal{S}}$ , except for node  $j$ , or indirectly observed nodes in  $\mathcal{V}$ . The result of the theorem implies that all parallel paths from  $w_i$  to  $w_{\mathcal{T}}$  and all loops around the ‘output’, i.e. all paths from  $w_j$  to  $w_{\mathcal{T}}$ , must pass through a node in this set.

**Remark 15.** The conditions in Theorem 14 are a generalization of the conditions for immersion. For the choice  $\mathcal{L} = \emptyset$  and  $\mathcal{V} = \emptyset$ , Algorithm 1 is equivalent to the immersion algorithm, and the results of Theorem 14 are equivalent to the conditions of Proposition 11, (Dankers et al., 2016). In the generalized situation, parallel paths  $w_i \rightarrow w_j$  and loops around the output  $w_j \rightarrow w_j$  can also be blocked by indirectly observed nodes, present in  $\mathcal{V}$ , instead of just by directly observed nodes in  $\tilde{\mathcal{S}}$ .

**Remark 16.** The conditions in Theorem 14 are a generalization of the conditions for the indirect inputs method, as formulated in Linder and Enqvist (2017b). This latter method results if we consider the particular situation that indirect observations are no in-neighbors of the output node, i.e.  $G_{j\mathcal{L}} = 0$ , and that all in-neighbors of indirect observations are in  $\mathcal{S} \cup \mathcal{V}$ , i.e.  $G_{\mathcal{L}\tilde{\mathcal{Z}}} = 0$ . In the generalized situation presented here, indirect observations  $w_{\mathcal{L}}$  are allowed to be in-neighbors of  $w_j$ , and they are allowed to have abstracted nodes  $w_{\mathcal{Z}}$  as in-neighbors.

## 7. Node selection strategy

Theorem 14 allows us to check whether a module remains invariant under abstraction if the network topology is known and we have divided the nodes into four groups. The next question is how to choose the sets of nodes, based on the network topology, such that the module of interest remains invariant.

### 7.1. Selecting the sets of nodes

The strategy to obtaining a set of measured nodes in Dankers et al. (2016) is as follows. First the input and output nodes of the module of interest are required to be measured. Then every parallel path from the input to the output node must be blocked by a measured node. This means that nodes are added such that each of those paths contains a measured node. Similarly every loop around the output node must be blocked by a measured node, so nodes are added such that each of those loops contains a measured node. Different nodes on a path can be chosen to block the path, so the choice of which nodes to measure is not unique.

Now, the method of choosing nodes is adapted with the possibility of using indirectly observed nodes. A parallel path or a loop can now be blocked by either a measured or an indirectly observed node. However, when we use an indirect observation to block a path, additional conditions must be satisfied. Paths from

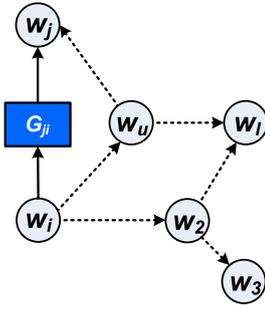


Fig. 7. Network where measured nodes and indirectly observed nodes are to be selected.

either input or output of the module of interest to the indirect observation must also be blocked by either a measured or an indirectly observed node. For each indirect observation that is added, this condition on blocking the paths is applied recursively. This selection method is demonstrated in the following example.

**Example 17 (Selecting Nodes).** For an illustration of how to select nodes, consider the network in Fig. 7. The module of interest is  $G_{ji}$ , so we select  $w_j$  as output, and  $w_i$  is included as a predictor input. A parallel path through node  $w_u$  exists and must be blocked if  $G_{ji}$  is to remain invariant. We can either include  $w_u$  as a predictor input, or we can choose to indirectly observe it using  $w_1$ . When  $w_1$  is chosen as indirect input measurement,  $l \in \mathcal{L}$ , and the parallel path from  $w_i$  to  $w_1$  through  $w_2$  should be blocked, so either  $w_2$  should then be included as a predictor input, or  $w_3$  can be included as the indirect observation of  $w_2$ .  $\square$

## 7.2. External excitation

In an estimation setting, both the nodes  $w_k, k \in \tilde{\mathcal{S}} \setminus j$  and  $w_l, l \in \mathcal{L}$  are used as predictor inputs to parameterized modules. The question can then be raised what the effect is on the identification setup for nodes being present in one of these sets. The external variables that are in-neighbors of the node  $w_j$  in the abstracted network need to be included as input. Depending on the chosen nodes in  $\tilde{\mathcal{S}}$  and  $\mathcal{L}$  different external variables are in-neighbors of the node  $w_j$ , so different external variables need to be chosen in the experimental setup. We have seen that placing a node in either  $\mathcal{L}$  or  $\tilde{\mathcal{S}}$  leads to a different transformation matrix  $P^{(abs)}$ . This leads to different locations of zeros in the transformed  $\check{R}$ . The structure of  $\check{R}$  can be described as follows. Let  $D$  denote the structure of a matrix that is diagonal, and let  $*$  denote a matrix of arbitrary structure, then

$$\check{u}_S = \begin{bmatrix} D & * & 0 & * \\ 0 & * & D & G_{\mathcal{L}\mathcal{V}}^{(1)} & * \end{bmatrix} R \begin{bmatrix} r_{\tilde{\mathcal{S}}} \\ r_{\mathcal{L}} \\ r_{\mathcal{V}} \\ r_{\tilde{\mathcal{Z}}} \end{bmatrix}, \quad (54)$$

where

$$G_{\mathcal{L}\mathcal{V}}^{(1)} = (G_{\mathcal{L}\mathcal{V}} + G_{\mathcal{L}\tilde{\mathcal{Z}}}(I - G_{\tilde{\mathcal{Z}}\tilde{\mathcal{Z}}})^{-1}G_{\tilde{\mathcal{Z}}\mathcal{V}}). \quad (55)$$

If we consider situation that the conditions of Proposition 10 are satisfied, e.g. where  $R$  and  $G_{\mathcal{L}\mathcal{V}}$  are diagonal, where  $G_{\mathcal{L}\tilde{\mathcal{Z}}} = 0$ ,  $G_{\mathcal{V}\mathcal{V}} = 0$ , and  $G_{\mathcal{V}\tilde{\mathcal{Z}}} = 0$ , and considering that  $D$  is diagonal, then the external excitations

$r_j$ ,  $r_{\mathcal{L}}$ , and  $r_{\tilde{\mathcal{Z}}}$

may be in-neighbor of  $w_j$ . In terms of choosing a network model set for the abstracted network, the structure of (54) specifies how to choose the zero-structure of the parameterized  $R(q, \theta)$  that is to be used for estimation of the abstracted network.

## 7.3. The noise model

Due to the abstraction, the noise process is modified in a way that is the same as the modification of the external excitations. The following expression is obtained for the disturbances

$$\check{v}_S = \begin{bmatrix} D & * & 0 & * \\ 0 & * & D & G_{\mathcal{L}\mathcal{V}}^{(1)} & * \end{bmatrix} H \begin{bmatrix} e_{\tilde{\mathcal{S}}} \\ e_{\mathcal{L}} \\ e_{\mathcal{V}} \\ e_{\tilde{\mathcal{Z}}} \end{bmatrix} \quad (56)$$

with  $G_{\mathcal{L}\mathcal{V}}^{(1)}$  specified in (55). The obtained noise filter above is not square, which is problematic in terms of identification. This noise model relates to a square noise filter  $\check{H}$  and white noise  $\check{e}$  that can be used in an identification setting. It is likely that the obtained  $\check{H}$  is then no longer diagonal. The zero-structure of the obtained noise filter can be used as the zero-structure when parameterizing the network model set. Under particular conditions special noise structures can be obtained that can be exploited. If no particular structure is obtained for the noise model, then all process noises are correlated.

## 7.4. Identification methods

For a particular network, and a choice of target module  $G_{ji}$ , the choice of the node sets  $\tilde{\mathcal{S}}, \mathcal{L}, \mathcal{V}, \tilde{\mathcal{Z}}$  will determine whether the target module will remain invariant in the abstracted network. This result can be applied in the problem of identifying the target module  $G_{ji}$  on the basis of measured node signals. In the abstracted network we have the node signals  $w_S$ , which we assume to be available from observations. We can now construct an identification setup in line with the methods developed in Van den Hof et al. (2013). Determine the set of input predictors as those node signals in  $w_S$  that are in-neighbors of the output  $w_j$  in the abstracted network. If  $G_{ji}$  has remained invariant in the abstracted network, i.e. when the conditions of Theorem 14 are satisfied, the identification problem of estimating the transfer functions from inputs in  $w_S$  to output  $w_j$  will now estimate the module from input  $w_i$  to output  $w_j$  that is equal to the module  $G_{ji}$  in the original network. Consistent identification of this module is then possible under the typical regularity conditions of the prediction error methods, as formulated in Van den Hof et al. (2013). This implies that:

- For the two-stage identification method, consistency of the estimate  $\hat{G}_{ji}$  is achievable if there is a sufficient excitation by external excitation signals in the network;
- For the direct identification method, consistency of the estimate  $\hat{G}_{ji}$  is achievable, if besides sufficient excitation by external excitation and disturbance signals, correlated noises between inputs and outputs are taken care of. This can be done by either choosing the node sets  $\tilde{\mathcal{S}}, \mathcal{L}, \mathcal{V}, \tilde{\mathcal{Z}}$  such that these correlated noises (or confounding variables) do not occur, Dankers et al. (2017), or by modeling this noise correlation correctly in the model, leading to the so-called joint-direct method as presented in Weerts et al. (2018b) and Van den Hof, Ramaswamy, Dankers, and Bottegal (2019).

When applied to the abstracted network, the prediction error associated with the joint-direct method is

$$\varepsilon(\theta) = \check{H}^{-1}(\theta) \left( (I - \check{G}(\theta))w_S - \check{R}(\theta)r \right), \quad (57)$$

where the  $\check{G}(\theta), \check{R}(\theta), \check{H}^{-1}(\theta)$  are structured according to the topology obtained by network abstraction, the  $w_S$  are all retained nodes, and  $r$  are all available external excitations. Then  $\varepsilon^T \varepsilon$  is minimized over the parameters to obtain the estimated model. A further analysis of the particular identification results is beyond the scope of this paper.

## 8. Conclusions

The question to be answered is which set of measured nodes can lead to consistent estimates of a target module. As a way to answer this question the concept of abstraction has been introduced as a way to remove unmeasured nodes from a network representation, as a generalization of methods present in literature. A systematic method has been introduced to select nodes such that the module of interest remains invariant in the abstracted network. Under some assumptions on external excitations and the network topology the abstracted network can be parameterized with a network identifiable model set. If the module of interest remains invariant, and the model set is identifiable, then conditions for consistent estimation can be obtained for various identification methods.

A requirement that has been imposed is that the module of interest remains invariant in the abstracted network, but this is not necessary for consistency. It may be that the module of interest can be identified in an indirect way by combining the knowledge of two or more modules present in the abstracted network. It is also possible that there are multiple sets of measured nodes that each lead to consistent estimates of a module of interest. Selecting the set of nodes that leads to the smallest variance is another question for future consideration.

## Acknowledgments

The authors are grateful for the discussions with Arne Dankers on the topic of the paper.

## Appendix

### A.1. Proof of Proposition 5

Sufficiency: By (15) the diagonal of  $G^{(2)}$  is

$$\text{diag}(G^{(2)}) = \text{diag}(I - P(I - G^{(1)})), \quad (58)$$

which is 0 by condition (2), showing that  $G^{(2)}$  is hollow. Moreover if Condition (1) is satisfied, then with (15), (16):

$$(I - G^{(2)})^{-1}R^{(2)} = (I - G^{(1)})^{-1}R^{(1)} \quad (59)$$

which is proper and stable by Definition 1. A monic, proper, stable, and inversely stable  $H^{(2)}$  and full rank  $\Lambda^{(2)}$  are obtained through the spectral factorization in (17).

Necessity: In order for  $(I - G^{(2)})^{-1}R^{(2)}$  to be proper and stable, it is required that  $P^{-1}$  exists. Therefore  $P$  has to have full rank. In order for  $G^{(2)}$  to be hollow, it is required that  $\text{diag}(I - P(I - G^{(1)})) = 0$ .  $\square$

### A.2. Proof of Proposition 6

Substituting  $P = (I - G^{(2)})(I - G^{(1)})^{-1}$  into the definition of the transformation (15) gives

$$G^{(2)} = I - (I - G^{(2)})(I - G^{(1)})^{-1}(I - G^{(1)}), \quad (60)$$

which shows that  $G^{(2)}$  is obtained by applying this transformation. Moreover the diagonal of  $(I - P(I - G^{(1)}))$  is 0, so  $P = (I - G^{(2)})(I - G^{(1)})^{-1}$  is an appropriate transformation.  $\square$

### A.3. Proof of Proposition 9

In order to prove the proposition we evaluate the expressions for each step of Algorithm 1.

**Step a:** The fourth equation of (27) is solved for  $w_{\bar{z}}$

$$w_{\bar{z}} = (I - G_{\bar{z}\bar{z}})^{-1} (G_{\bar{z}\bar{s}}w_{\bar{s}} + G_{\bar{z}\bar{L}}w_{\bar{L}} + G_{\bar{z}\bar{V}}w_{\bar{V}} + u_{\bar{z}} + v_{\bar{z}}). \quad (61)$$

Substituting the above equation into the remainder of the network results for  $w_{\bar{s}}$  in

$$w_{\bar{s}} = G_{\bar{s}\bar{s}}^{(1)}w_{\bar{s}} + G_{\bar{s}\bar{L}}^{(1)}w_{\bar{L}} + G_{\bar{s}\bar{V}}^{(1)}w_{\bar{V}} + u_{\bar{s}}^{(1)} + v_{\bar{s}}^{(1)}, \quad (62)$$

with

$$G_{\bar{s}\bar{s}}^{(1)} = (G_{\bar{s}\bar{s}} + G_{\bar{s}\bar{z}}(I - G_{\bar{z}\bar{z}})^{-1}G_{\bar{z}\bar{s}})$$

$$G_{\bar{s}\bar{L}}^{(1)} = (G_{\bar{s}\bar{L}} + G_{\bar{s}\bar{z}}(I - G_{\bar{z}\bar{z}})^{-1}G_{\bar{z}\bar{L}})$$

$$G_{\bar{s}\bar{V}}^{(1)} = (G_{\bar{s}\bar{V}} + G_{\bar{s}\bar{z}}(I - G_{\bar{z}\bar{z}})^{-1}G_{\bar{z}\bar{V}})$$

$$u_{\bar{s}}^{(1)} = u_{\bar{s}} + G_{\bar{s}\bar{z}}(I - G_{\bar{z}\bar{z}})^{-1}u_{\bar{z}}$$

$$v_{\bar{s}}^{(1)} = v_{\bar{s}} + G_{\bar{s}\bar{z}}(I - G_{\bar{z}\bar{z}})^{-1}v_{\bar{z}}.$$

For  $w_{\bar{L}}$  we obtain

$$w_{\bar{L}} = G_{\bar{L}\bar{s}}^{(1)}w_{\bar{s}} + G_{\bar{L}\bar{L}}^{(1)}w_{\bar{L}} + G_{\bar{L}\bar{V}}^{(1)}w_{\bar{V}} + u_{\bar{L}}^{(1)} + v_{\bar{L}}^{(1)}, \quad (63)$$

with

$$G_{\bar{L}\bar{s}}^{(1)} = (G_{\bar{L}\bar{s}} + G_{\bar{L}\bar{z}}(I - G_{\bar{z}\bar{z}})^{-1}G_{\bar{z}\bar{s}})$$

$$G_{\bar{L}\bar{L}}^{(1)} = (G_{\bar{L}\bar{L}} + G_{\bar{L}\bar{z}}(I - G_{\bar{z}\bar{z}})^{-1}G_{\bar{z}\bar{L}})$$

$$G_{\bar{L}\bar{V}}^{(1)} = (G_{\bar{L}\bar{V}} + G_{\bar{L}\bar{z}}(I - G_{\bar{z}\bar{z}})^{-1}G_{\bar{z}\bar{V}})$$

$$u_{\bar{L}}^{(1)} = u_{\bar{L}} + G_{\bar{L}\bar{z}}(I - G_{\bar{z}\bar{z}})^{-1}u_{\bar{z}}$$

$$v_{\bar{L}}^{(1)} = v_{\bar{L}} + G_{\bar{L}\bar{z}}(I - G_{\bar{z}\bar{z}})^{-1}v_{\bar{z}}.$$

For  $w_{\bar{V}}$  we obtain

$$w_{\bar{V}} = G_{\bar{V}\bar{s}}^{(1)}w_{\bar{s}} + G_{\bar{V}\bar{L}}^{(1)}w_{\bar{L}} + G_{\bar{V}\bar{V}}^{(1)}w_{\bar{V}} + u_{\bar{V}}^{(1)} + v_{\bar{V}}^{(1)}, \quad (64)$$

with

$$G_{\bar{V}\bar{s}}^{(1)} = (G_{\bar{V}\bar{s}} + G_{\bar{V}\bar{z}}(I - G_{\bar{z}\bar{z}})^{-1}G_{\bar{z}\bar{s}})$$

$$G_{\bar{V}\bar{L}}^{(1)} = (G_{\bar{V}\bar{L}} + G_{\bar{V}\bar{z}}(I - G_{\bar{z}\bar{z}})^{-1}G_{\bar{z}\bar{L}})$$

$$G_{\bar{V}\bar{V}}^{(1)} = (G_{\bar{V}\bar{V}} + G_{\bar{V}\bar{z}}(I - G_{\bar{z}\bar{z}})^{-1}G_{\bar{z}\bar{V}})$$

$$u_{\bar{V}}^{(1)} = u_{\bar{V}} + G_{\bar{V}\bar{z}}(I - G_{\bar{z}\bar{z}})^{-1}u_{\bar{z}}$$

$$v_{\bar{V}}^{(1)} = v_{\bar{V}} + G_{\bar{V}\bar{z}}(I - G_{\bar{z}\bar{z}})^{-1}v_{\bar{z}}.$$

It is straightforward to verify that the transformation  $G^{(1)} = I - P^{(1)}(I - G)$ ,  $u^{(1)} = P^{(1)}u$ , and  $v^{(1)} = P^{(1)}v$  results in the same expressions as the algorithm.

**Steps b and c:** The two equations that are solved for  $w_{\bar{V}}$  result in

$$w_{\bar{V}} = (G_{\bar{L}\bar{V}}^{(1)})^\dagger \left( -G_{\bar{L}\bar{s}}^{(1)}w_{\bar{s}} + (I - G_{\bar{L}\bar{L}}^{(1)})w_{\bar{L}} - u_{\bar{L}}^{(1)} - v_{\bar{L}}^{(1)} \right), \quad (65)$$

and

$$w_{\bar{V}} = (I - G_{\bar{V}\bar{V}}^{(1)})^{-1} \left( G_{\bar{V}\bar{s}}^{(1)}w_{\bar{s}} + G_{\bar{V}\bar{L}}^{(1)}w_{\bar{L}} + u_{\bar{V}}^{(1)} + v_{\bar{V}}^{(1)} \right). \quad (66)$$

Substituting (65) into (62) results in

$$w_{\bar{s}} = G_{\bar{s}\bar{s}}^{(3)}w_{\bar{s}} + G_{\bar{s}\bar{L}}^{(3)}w_{\bar{L}} + u_{\bar{s}}^{(3)} + v_{\bar{s}}^{(3)}, \quad (67)$$

with

$$G_{\bar{s}\bar{s}}^{(3)} = G_{\bar{s}\bar{s}}^{(1)} - G_{\bar{s}\bar{V}}^{(1)}(G_{\bar{L}\bar{V}}^{(1)})^\dagger G_{\bar{L}\bar{s}}^{(1)}$$

$$G_{\bar{s}\bar{L}}^{(3)} = G_{\bar{s}\bar{L}}^{(1)} + G_{\bar{s}\bar{V}}^{(1)}(G_{\bar{L}\bar{V}}^{(1)})^\dagger (I - G_{\bar{L}\bar{L}}^{(1)})$$

$$u_{\bar{s}}^{(3)} = u_{\bar{s}}^{(1)} + G_{\bar{s}\bar{V}}^{(1)}(G_{\bar{L}\bar{V}}^{(1)})^\dagger u_{\bar{L}}^{(1)}$$

$$v_{\mathcal{S}}^{(3)} = v_{\mathcal{S}}^{(1)} + G_{\mathcal{S}\mathcal{V}}^{(1)}(G_{\mathcal{L}\mathcal{V}}^{(1)})^\dagger v_{\mathcal{L}}^{(1)}.$$

Substituting (65) into (63) results in

$$w_{\mathcal{L}} = G_{\mathcal{L}\mathcal{S}}^{(3)} w_{\mathcal{S}} + G_{\mathcal{L}\mathcal{L}}^{(3)} w_{\mathcal{L}} + u_{\mathcal{L}}^{(3)} + v_{\mathcal{L}}^{(3)}, \quad (68)$$

with

$$G_{\mathcal{L}\mathcal{S}}^{(3)} = G_{\mathcal{L}\mathcal{S}}^{(1)} + G_{\mathcal{L}\mathcal{V}}^{(1)}(I - G_{\mathcal{V}\mathcal{V}}^{(1)})^{-1} G_{\mathcal{V}\mathcal{S}}^{(1)}$$

$$G_{\mathcal{L}\mathcal{L}}^{(3)} = G_{\mathcal{L}\mathcal{L}}^{(1)} + G_{\mathcal{L}\mathcal{V}}^{(1)}(I - G_{\mathcal{V}\mathcal{V}}^{(1)})^{-1} G_{\mathcal{V}\mathcal{L}}^{(1)}$$

$$u_{\mathcal{L}}^{(3)} = u_{\mathcal{L}}^{(1)} + G_{\mathcal{L}\mathcal{V}}^{(1)}(I - G_{\mathcal{V}\mathcal{V}}^{(1)})^{-1} u_{\mathcal{V}}^{(1)}$$

$$v_{\mathcal{L}}^{(3)} = v_{\mathcal{L}}^{(1)} + G_{\mathcal{L}\mathcal{V}}^{(1)}(I - G_{\mathcal{V}\mathcal{V}}^{(1)})^{-1} v_{\mathcal{V}}^{(1)}.$$

The combination of the transformations  $P^{(3)}P^{(2)}$  is denoted by

$$P^{(3,2)} := \begin{bmatrix} I & G_{\mathcal{S}\mathcal{V}}^{(2)}(G_{\mathcal{L}\mathcal{V}}^{(1)})^\dagger & 0 & 0 \\ 0 & I & G_{\mathcal{L}\mathcal{V}}^{(1)}(I - G_{\mathcal{V}\mathcal{V}}^{(1)})^{-1} & 0 \\ 0 & (G_{\mathcal{L}\mathcal{V}}^{(1)})^\dagger & 0 & 0 \\ 0 & G_{\mathcal{Z}\mathcal{V}}^{(2)}(G_{\mathcal{L}\mathcal{V}}^{(1)})^\dagger & 0 & I \end{bmatrix},$$

and it is straightforward to verify that this transformation  $G^{(3)} = I - P^{(3,2)}(I - G^{(1)})$ ,  $u^{(3)} = P^{(3,2)}u^{(1)}$ , and  $v^{(3)} = P^{(3,2)}v^{(1)}$  results in the same expressions as the algorithm.

**Step d:** Removal of a self-loop in the equation for node  $w_j$  is performed by subtracting the right-hand term for  $w_j$  from both sides, and then dividing both sides by 1 minus the term. This is precisely the operation performed by the transformation  $P^{(4)}$ .  $\square$

#### A.4. Proof of Proposition 10

Transfer function matrix  $\check{R}$  is

$$\check{R} = \begin{bmatrix} I & 0 \end{bmatrix} P^{(4)} P^{(3)} P^{(2)} P^{(1)} R, \quad (69)$$

where the transformations are defined in Section 4.2. The structural properties of the matrices will be evaluated. Let  $*$  indicate an unstructured matrix, and let  $D$  indicate a diagonal matrix structure. Then the structure of the first part of the transformation matrix is

$$\begin{bmatrix} I & 0 \end{bmatrix} P^{(4)} P^{(3)} = \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} D & * \\ 0 & * \end{bmatrix} = \begin{bmatrix} D & * \end{bmatrix}, \quad (70)$$

where the matrices are partitioned corresponding to the blocks  $\mathcal{S} = \mathcal{S} \cup \mathcal{L}$  and  $\mathcal{Z} = \mathcal{Z} \cup \mathcal{V}$ .

The transformation  $P^{(2)}P^{(1)}$  has the following structure

$$P^{(2)}P^{(1)} = \begin{bmatrix} D & 0 & 0 & 0 \\ 0 & D & X & 0 \\ 0 & * & 0 & 0 \\ 0 & 0 & 0 & D \end{bmatrix} \begin{bmatrix} D & 0 & 0 & * \\ 0 & D & 0 & * \\ 0 & 0 & D & * \\ 0 & 0 & 0 & * \end{bmatrix} \quad (71)$$

$$= \begin{bmatrix} D & 0 & 0 & * \\ 0 & D & X & * \\ 0 & * & 0 & * \\ 0 & 0 & 0 & * \end{bmatrix},$$

where the matrices are partitioned corresponding to the blocks  $\mathcal{S}$ ,  $\mathcal{L}$ ,  $\mathcal{V}$ ,  $\mathcal{Z}$ , and where  $X = G_{\mathcal{L}\mathcal{V}}^{(1)}(I - G_{\mathcal{V}\mathcal{V}}^{(1)})^{-1}$ . From the relations in (63) and (64) we obtain that

$$G_{\mathcal{L}\mathcal{V}}^{(1)} = (G_{\mathcal{L}\mathcal{V}} + G_{\mathcal{L}\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1} G_{\mathcal{Z}\mathcal{V}})$$

and

$$G_{\mathcal{V}\mathcal{V}}^{(1)} = (G_{\mathcal{V}\mathcal{V}} + G_{\mathcal{V}\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1} G_{\mathcal{Z}\mathcal{V}}).$$

Then from condition 3 we obtain that  $X$  is diagonal.

Under conditions 1 and 2, matrix  $R$  has the structure

$$R = \begin{bmatrix} D & * & 0 & * \\ 0 & * & 0 & * \\ 0 & * & D & * \\ 0 & * & 0 & * \end{bmatrix}, \quad (72)$$

such that the following is obtained

$$P^{(2)}P^{(1)}R = \begin{bmatrix} D & * & 0 & * \\ 0 & * & D & * \\ 0 & * & 0 & * \\ 0 & * & 0 & * \end{bmatrix}. \quad (73)$$

Then the final structure is

$$\check{R} = \begin{bmatrix} D & 0 & * & * \\ 0 & D & * & * \end{bmatrix} \begin{bmatrix} D & * & 0 & * \\ 0 & * & D & * \\ 0 & * & 0 & * \\ 0 & * & 0 & * \end{bmatrix} = \begin{bmatrix} D & * & 0 & * \\ 0 & * & D & * \end{bmatrix}. \quad (74)$$

It is then obvious that a leading diagonal can be obtained by column operations.  $\square$

#### A.5. Proof of Theorem 14

In order to prove the theorem, the conditions must be interpreted in terms of  $G$ . Conditions (a) and (b) imply that there are no direct paths from  $w_i$  and  $w_j$  to indirect observations  $w_{\mathcal{L}}$ , i.e.

- (i)  $G_{\mathcal{L}i} = 0$ ,
- (ii)  $G_{\mathcal{L}j} = 0$ .

The conditions also imply that there are no paths from  $w_i$  and  $w_j$  to indirect observations  $w_{\mathcal{L}}$  and  $j$  that only go through unmeasured nodes  $w_{\mathcal{Z}}$ , i.e.

- (iii)  $G_{\mathcal{L}\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1} G_{\mathcal{Z}i} = 0$ ,
- (iv)  $G_{\mathcal{L}\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1} G_{\mathcal{Z}j} = 0$ ,
- (v)  $G_{j\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1} G_{\mathcal{Z}i} = 0$ ,
- (vi)  $G_{j\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1} G_{\mathcal{Z}j} = 0$ .

The module of interest is a part of  $G^{(4)}$  in (37) which can be obtained as

$$G^{(4)} = I - P^{(4)}(I - G^{(3)}). \quad (75)$$

Explicit expressions can be found in the proof of Proposition 9 such that using (i)–(vi) we can see that  $G_{ji}^{(4)} = G_{ji}$ . First it is shown that  $G_{ji}^{(3)} = G_{ji}$ . From (67) we obtain that

$$G_{ji}^{(3)} = G_{ji}^{(1)} - G_{j\mathcal{V}}^{(1)}(G_{\mathcal{L}\mathcal{V}}^{(1)})^\dagger G_{\mathcal{L}i}^{(1)}. \quad (76)$$

Then it can be observed that

$$G_{\mathcal{L}i}^{(1)} = G_{\mathcal{L}i} + G_{\mathcal{L}\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1} G_{\mathcal{Z}i} = 0 \quad (77)$$

if we use the expression from (63) and the obtained conditions (i) and (iii). Now  $G_{ji}^{(3)} = G_{ji}^{(1)}$  is evaluated using the expression in (62)

$$G_{ji}^{(3)} = G_{ji} + G_{j\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1} G_{\mathcal{Z}i}. \quad (78)$$

Then by condition (v) we have  $G_{ji}^{(3)} = G_{ji}$ .

Since  $P^{(4)}$  is diagonal, all that is left to show is that its  $jj$ th entry is 1. Using the expression (35) we then need to show that  $G_{jj}^{(3)} = 0$ . From (67) we obtain that

$$G_{jj}^{(3)} = G_{jj}^{(1)} - G_{j\mathcal{V}}^{(1)}(G_{\mathcal{L}\mathcal{V}}^{(1)})^\dagger G_{\mathcal{L}j}^{(1)}. \quad (79)$$

Then it can be observed that

$$G_{\mathcal{L}j}^{(1)} = G_{\mathcal{L}j} + G_{\mathcal{L}\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1} G_{\mathcal{Z}j} = 0 \quad (80)$$

if we use the expression from (63) and the obtained conditions (ii) and (iv). Now  $G_{jj}^{(1)}$  is evaluated using the expression in (62)

$$G_{jj}^{(1)} = G_{jj} + G_{j\mathcal{Z}}(I - G_{\mathcal{Z}\mathcal{Z}})^{-1} G_{\mathcal{Z}j}. \quad (81)$$

Since there are no self loops in  $G$ , the  $G_{jj} = 0$ . Then by condition (vi) we have  $G_{jj}^{(3)} = 0$ , such that  $P_{jj}^{(4)} = 1$  and  $G_{ji}^{(4)} = G_{ji}$ .  $\square$

## References

- Bazanella, A. S., Gevers, M., Hendrickx, J., & Parraga, A. (2017). Identifiability of dynamical networks: which nodes need be measured? In *Proc. 56th IEEE conference on decision and control* (pp. 5870–5875).
- Chiuseo, A., & Pilonetto, G. (2012). A Bayesian approach to sparse dynamic network identification. *Automatica*, 48(8), 1553–1565.
- Dankers, A. G. (2014). *System identification in dynamic networks* (Ph.D. thesis), Delft University of Technology.
- Dankers, A. G., Van den Hof, P. M. J., Bombois, X., & Heuberger, P. S. C. (2015). Errors-in-variables identification in dynamic networks—Consistency results for an instrumental variable approach. *Automatica*, 62, 39–50.
- Dankers, A. G., Van den Hof, P. M. J., Heuberger, P. S. C., & Bombois, X. (2016). Identification of dynamic models in complex networks with prediction error methods: Predictor input selection. *IEEE Transactions on Automatic Control*, 61(4), 937–952.
- Dankers, A. G., Van den Hof, P. M. J., Materassi, D., & Weerts, H. H. M. (2017). Conditions for handling confounding variables in dynamic networks. *IFAC-PapersOnLine*, 50-1, 3983–3988, Proc. 20th IFAC world congress.
- Everitt, N., Bottegal, G., & Hjalmarsson, H. (2018). An empirical Bayes approach to identification of modules in dynamic networks. *Automatica*, 91, 144–151.
- Gevers, M., Bazanella, A., & Vian da Silva, G. (2018). A practical method for the consistent identification of a module in a dynamical network. *IFAC-PapersOnLine*, 51-15, 862–867, Proc. 18th IFAC Symp. System Identif. (SYSID2018).
- Gonçalves, J., & Warnick, S. (2008). Necessary and sufficient conditions for dynamical structure reconstruction of LTI networks. *IEEE Transactions on Automatic Control*, 53(7), 1670–1674.
- Hendrickx, J., Gevers, M., & Bazanella, A. (2019). Identifiability of dynamical networks with partial node measurements. *IEEE Transactions on Automatic Control*, 64(6), 2240–2253.
- Linder, J. (2017). *Indirect system identification for unknown input problems with applications to ships* (Ph.D. thesis), Linköping University.
- Linder, J., & Enqvist, M. (2017a). Identification and prediction in dynamic networks with unobservable nodes. *IFAC-PapersOnLine*, 50(1), 10574–10579, Proc. 20th IFAC world congress.
- Linder, J., & Enqvist, M. (2017b). Identification of systems with unknown inputs using indirect input measurements. *International Journal of Control*, 90(4), 729–745.
- Materassi, D., & Salapaka, M. (2012). On the problem of reconstructing an unknown topology via locality properties of the Wiener filter. *IEEE Transactions on Automatic Control*, 57(7), 1765–1777.
- Materassi, D., & Salapaka, M. V. (2015). Identification of network components in presence of unobserved nodes. In *Proc. 54th IEEE conference on decision and control* (pp. 1563–1568).
- Materassi, D., & Salapaka, M. V. (2019). Signal selection for estimation and identification in networks of dynamic systems: a graphical model approach. ArXiv preprint arXiv:1905.12132.
- Pappas, G. J., & Sastry, S. (1997). Hybrid systems IV. In P. Antsaklis, W. Kophn, A. Nerode, & S. Sastry (Eds.), *Lecture notes in computer science: Vol. 1273*, (pp. 329–341). Springer.
- Ramaswamy, K. R., Bottegal, G., & Van den Hof, P. M. J. (2018). Local module identification in dynamic networks using regularized kernel-based methods. In *Proc. 57th IEEE conference on decision and control* (pp. 4714–4719).
- Sanandaji, B. M., Vincent, T. L., & Wakin, M. B. (2011). Exact Topology Identification of Large-Scale Interconnected Dynamical Systems from Compressive Observations. In *Proc. American control conference* (pp. 649–656). San Francisco, CA, USA.
- Van den Hof, P. M. J., Dankers, A. G., Heuberger, P. S. C., & Bombois, X. (2013). Identification of dynamic models in complex networks with prediction error methods - basic methods for consistent module estimates. *Automatica*, 49(10), 2994–3006.
- Van den Hof, P. M. J., Ramaswamy, K. R., Dankers, A. G., & Bottegal, G. (2019). Local module identification in dynamic networks with correlated noise: the full input case. In *Proc. 57th IEEE conf. decision and control* (pp. 5494–5499).
- van der Woude, J. (1991). A graph-theoretic characterization for the rank of the transfer matrix of a structured system. *Mathematics of Control, Signals, and Systems*, 4, 33–40.
- Weerts, H. H. M., Van den Hof, P. M. J., & Dankers, A. G. (2018a). Identifiability of linear dynamic networks. *Automatica*, 89, 247–258.
- Weerts, H. H. M., Van den Hof, P. M. J., & Dankers, A. G. (2018b). Identification of linear dynamic networks with rank-reduced noise. *Automatica*, 98, 256–268.

Weerts, H. H. M., Van den Hof, P. M. J., & Dankers, A. G. (2018c). Single module identifiability in linear dynamic networks. In *Proc. 57th IEEE conference on decision and control* (pp. 4725–4730).

Woodbury, N., Dankers, A., & Warnick, S. (2017). On the well-posedness of lti networks. In *Proc. 56th IEEE conference on decision and control* (pp. 4813–4818).

Yuan, Y., Stan, G. -B., Warnick, S., & Gonçalves, J. (2011). Robust dynamical network structure reconstruction. *Automatica*, 47(6), 1230–1235.

Zorzi, M., & Chiuseo, A. (2017). Sparse plus low rank network identification: a nonparametric approach. *Automatica*, 76, 355–366.



**Harm H.M. Weerts** received his B.Sc. degree from the Fontys University of Applied Sciences in Venlo, The Netherlands, in 2010, and his M.Sc. degree in Systems & Control from Eindhoven University of Technology, The Netherlands, in 2014. In 2018 he obtained the Ph.D. degree at Eindhoven University of Technology on the topic of system identification applied to dynamic networks with a multivariable approach. Currently he is employed at ALTEN Nederland as a mechatronics consultant.



**Jonas Linder** received the B.Sc. and M.Sc. degrees in systems, control and mechatronics from Chalmers University of Technology in Gothenburg, Sweden, in 2009 and 2011, respectively. In 2017, he received his Ph.D. degree in automatic control from the department of electrical engineering at Linköping University, Sweden, and continued as a postdoctoral researcher in collaboration with ABB Corporate Research in Västerås, Sweden, and ABB Marine in Helsinki, Finland. In 2018, he obtained a full-time position at ABB Corporate Research where he applies control theory on real-world emphasis on data-based modeling, sensor fusion and

problems with current simulation methods.



**Martin Enqvist** is an associate professor of automatic control at Linköping University, Sweden, since 2008. He was a research associate at the same university from 2007 to 2008 and a postdoctoral researcher at Vrije Universiteit Brussel, Belgium, during 2006. He obtained the M.Sc. degree in applied physics and electrical engineering in 2000 and the Ph.D. degree in automatic control in 2005, both from Linköping University. His main research interests are within the field of system identification, with a particular focus on nonlinear systems and approximation issues, and he has worked

with applications in aeronautics, marine automation and communication systems. Furthermore, he was an associate editor for *Automatica* between 2008 and 2019.



**Paul M. J. Van den Hof** received the M.Sc. and Ph.D. degrees in electrical engineering from Eindhoven University of Technology, Eindhoven, The Netherlands, in 1982 and 1989, respectively. In 1986 he moved to Delft University of Technology, where he was appointed as Full Professor in 1999. From 2003 to 2011, he was founding co-director of the Delft Center for Systems and Control (DCSC). As of 2011, he is a Full Professor in the Electrical Engineering Department, Eindhoven University of Technology. His research interests include system identification, identification for control, and model-based control and optimization, with applications in industrial process control systems and high-tech systems. He holds an ERC Advanced Research grant for a research project on identification in dynamic networks. Paul Van den Hof is an IFAC Fellow and IEEE Fellow, and Honorary Member of the Hungarian Academy of Sciences. He has been a member of the IFAC Council (1999–2005, 2017–2020), the Board of Governors of IEEE Control Systems Society (2003–2005), and an Associate Editor and Editor of *Automatica* (1992–2005). In the triennium 2017–2020 he served as Vice-President of IFAC.