To appear in Proc. 18th IFAC Symposium on System Identification (SYSID 2018), 9-11 July 2018, Stockholm, Sweden.

On Representations of Linear Dynamic Networks ⋆

E.M.M. Kivits * and **Paul M.J. Van den Hof** *

* *Control Systems Group, Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands (email: e.m.m.kivits@tue.nl, p.m.j.vandenhof@tue.nl)*

Abstract: Linear dynamic networks are typically described in either a state-space form or a module representation. The question is addressed under which conditions these representations are equivalent and can be transformed into one another. Hidden states and especially shared hidden states have a central position in this analysis. A consequence for identification is that MIMO parameterised modules may be necessary in order to appropriately take care of shared hidden states. Further, the construction of sub-networks in a linear dynamic network resulting in a module representation is illustrated. The module dynamic network allows to zoom in/out on/of the network to include/exclude more detailed structural information. Zooming in and out is respectively described by realisation and multi-path immersion.

Keywords: System identification, dynamic networks

1. INTRODUCTION

Linear dynamic networks are interconnections of linear dynamic systems. The attention for dynamic networks is growing, because in current day's technology, systems are increasing in complexity and size and an increasing number of systems is being interconnected. The interest in identification, control and reduction of dynamic networks is spreading over a diversity of scientific fields such as social science, finance, computer science, bio-informatics, biology and engineering. As a result, a variety of representations of dynamic networks is developed and the question arises how these representations are related to each other.

In one part of the literature, state-space forms are used as a basis of dynamic network descriptions. State-space forms are typically related to first principles modelling and can be very much appealing in this sense. State-space descriptions can be depicted in several ways. Often only the structure of the network is drawn in a directed or undirected graph, where the nodes are the states of the system, see e.g. Materassi and Salapaka (2015) and their references.

Sometimes, the edges of the graph are weighted with the corresponding elements of the system matrices, as in Chang et al. (2014), which gives some insight in the relations between the inputs, states and outputs. The weights can also be dynamic transfer functions, which closely relates to the dynamical structure function described by Gonçalves et al. (2007) and the module representation of Van den Hof et al. (2013).

⋆ 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).

A dynamic network formulation in an identification context has been introduced in Van den Hof et al. (2013). Dynamic networks are considered in a node and link structure, including noise disturbances, excitation signals and sensor noises, see Dankers et al. (2015). The network is based on scalar transfer function links (modules) between node signals.

Some different representations of dynamic networks are presented in Yeung et al. (2010), Yeung et al. (2011), Chetty and Warnick (2015) and Warnick (2015). They characterise the structure of dynamic networks on different levels. The emphasis is on the difference between the dynamical structure function and the module representation, while the relation to state-space forms is given less attention.

In terms of identification in dynamic networks, the following problems have been addressed so far: the identification of a single module: e.g. Van den Hof et al. (2013), Materassi and Salapaka (2015), Dankers et al. (2016); the identification of all modules: e.g. Risuleo et al. (2017); the identification of the structure or topology: e.g. Materassi and Innocenti (2010); and the identifiability of the network: e.g. Weerts et al. (2018).

The main question in this paper is: can a state-space form always be converted into a module representation without losing any information and vice versa? To answer this question, algorithms are developed to transform one network representation into the other. The focus is on discrete time systems, although the results are applicable to continuous time systems as well.

The paper is organised as follows. Section 2 defines the module dynamic network and the state-space dynamic network. The relations between these two representations are described in Section 3. Section 4 extends to more general networks. Section 5 describes the division of a net-

work into sub-networks. Section 6 contains the discussion and Section 7 presents the conclusion. The proofs of the lemmas and propositions are included in a report version of the paper: Kivits and Van den Hof (2018).

2. REPRESENTATIONS OF DYNAMIC NETWORKS

2.1 Module dynamic networks

A module representation of dynamic networks as considered in this paper is based on Van den Hof et al. (2013). A dynamic network is the interconnection of L nodes $w_j(t)$, $j = 1, \ldots, L$, and K known external excitation signals $r_k(t)$, $k = 1, \ldots, K$. Each node signal is equal to

$$w_j(t) = \sum_{i=1}^{L} G_{ji}(q) w_i(t) + \sum_{k=1}^{K} R_{jk}(q) r_k(t), \qquad (1)$$

where $G_{ji}(q)$ and $R_{jk}(q)$ are proper rational transfer functions with q^{-1} the delay operator meaning $q^{-1}w_j(t) = w_j(t-1)$. As a further generalisation of the setup in Van den Hof et al. (2013), the signals $w_j(t)$ and $r_k(t)$ can be vector-valued in which case the related transfer functions become matrices of appropriate dimensions; additionally self-loops are allowed, i.e. $G_{ii}(q)$ is not necessarily 0.

Typically, node signals are affected by unknown disturbance signals. In this paper, unknown inputs act similar to known inputs and therefore disturbances are initially omitted for simplicity and considered in Section 4.

The expressions for the node signals (1) can be combined in a matrix equation describing the network as

$$w(t) = Gw(t) + Rr(t), \qquad (2)$$

$$w(t) = (I - G)^{-1} Rr(t), \qquad (3)$$

with matrices G and R composed of elements $G_{ji}(q)$ and $R_{jk}(q)$ respectively, and where $w(t)$ and $r(t)$ are vectorised versions of $w_j(t)$ and $r_k(t)$ respectively. All minors of $I - G(\infty)$ should be non-zero in order to achieve a well-posed network. Equation (2) is a dynamical structure function as introduced by Gonçalves et al. (2007). Figure 1a shows a single building block of a module dynamic network.

Definition 1. (Module dynamic network). A module dynamic network is defined by the pair (G, R) describing a map $r(t) \to w(t)$ according to (3).

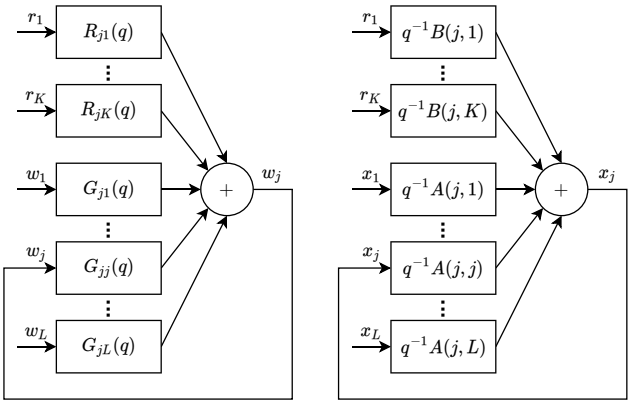
Definition 2. (Equal and equivalent networks). Consider the networks $\mathcal{N}_1$ and $\mathcal{N}_2$ defined by the pairs (G_1, R_1) and (G_2, R_2) respectively. The networks $\mathcal{N}_1$ and $\mathcal{N}_2$ are

- equal if $(G_1, R_1) = (G_2, R_2)$.
- equivalent if $(I - G_1)^{-1} R_1 = (I - G_2)^{-1} R_2$.

A particular property of the module representations in literature is that they only allow for SISO modules and exclude self-loops. This choice has been made to avoid identifiability problems as implied by the following lemma.

Lemma 1. (Self-loops). A module dynamic network with self-loops can always equivalently be written as a module dynamic network without self-loops.

In this paper, self-loops are allowed in module dynamic networks in order to be able to link with state-space dynamic networks as described in the next section.



(a) A single node of a module dynamic network.

(b) A single node of an SS dynamic network.

Fig. 1. A single node of a module and SS dynamic network.

Network properties that depend on the structure or topology are referred to as structural or generic network properties. One such property is the *generic McMillan degree*, see Karcanias et al. (2005).

Definition 3. (Generic McMillan degree). Let a system be represented by a coefficient vector $\theta \in \mathbb{R}^{n_\theta}$. The generic McMillan degree of the system is the McMillan degree of the system for almost all θ, i.e. for all $\theta \in \mathbb{R}^{n_\theta}$ except for a set of measure 0.

Lemma 2. (Generic McMillan degree). The generic McMillan degree of a network is equal to the sum of the generic McMillan degrees of all modules in the network.

2.2 State-space (SS) dynamic networks

In some research areas the behaviour of a dynamic network is mathematically described in a state-space form as

$$x(t+1) = Ax(t) + Br(t), \qquad (4)$$

where $x(t)$ is the state variable and $r(t)$ is a known external excitation signal. A state-space description can also be depicted as a module dynamic network. Figure 1b shows a single building block of a state-space dynamic network.

Definition 4. (State-space (SS) dynamic network). A state-space (SS) dynamic network is a module dynamic network, as defined in Definition 1, with the additional properties that

- Every state variable $x_j(t)$ is a node signal $w_j(t)$.
- Every element in G has the form $G_{ji}(q) = q^{-1}A(j,i)$.
- Every element in R has the form $R_{jk}(q) = q^{-1}B(j,k)$.

In general, SS dynamic networks contain self-loops, because $A(i,i) \neq 0$. These diagonal elements of A represent the relation from $x_i(t)$ to $x_i(t+1)$.

3. RELATIONS BETWEEN MODULE AND SS DYNAMIC NETWORKS

3.1 Abstraction of state-space dynamic networks

One of the major expansions of module dynamic networks compared to SS dynamic networks is that in module dynamic networks the states are grouped into a single module, while in SS dynamic networks the network is split into

its core elements with modules that only have (weighted) delays. A natural step to go from SS dynamic networks to general module dynamic networks is by grouping states, that is, by removing state variables as node signals and thereby increasing the order of the dynamic terms in the modules of the network. This process is referred to as *abstraction*, see Woodbury et al. (2017).

Definition 5. (Abstraction). Consider the networks $\mathcal{N}_1$ and $\mathcal{N}_2$ defined by the pairs (G_1, R_1) and (G_2, R_2) respectively. $\mathcal{N}_1$ is an abstraction of $\mathcal{N}_2$ with respect to nodes $w_\alpha(t)$ if $(I - G_1)^{-1}R_1 = \left[(I - G_2)^{-1}R_2\right]_{w_\alpha}$, where $[T]_{w_\alpha}$ means T without the rows corresponding to nodes $w_\alpha(t)$.

Abstracted node signals are still present in the network, but are hidden in the modules and therefore referred to as *hidden states*. Hidden states present in multiple modules are said to be shared by these modules and therefore referred to as *shared hidden states*, see Warnick (2015).

Abstraction is performed by eliminating $w_\alpha(t)$ from the node equations and is non-unique, since various node equations can be used for this, Weerts and Linder (2018). If the node equation of $w_\alpha(t)$ itself is used, this approach is referred to as *immersion* and has been worked out in Dankers et al. (2016) for module dynamic networks without self-loops and with only SISO modules, and is equivalent to vertex elimination. This immersion process is generalised to dynamic networks with self-loops in the following algorithm and is equivalent to the Kron reduction, see Dörfler and Bullo (2013).

Algorithm 1. (Immersion). An abstraction of a module dynamic network is obtained through immersion by taking the following steps:

(1) Select a node $w_\alpha(t)$.
(2) Substitute the node equation of $w_\alpha(t)$ into the other node equations.
(3) Delete the node equation of $w_\alpha(t)$ from the network.
(4) Repeat the procedure to abstract more nodes.

Graphically, immersion is performed by lifting the paths through $w_\alpha(t)$ and deleting the isolated $w_\alpha(t)$.

Example 1. (Lifting path). Consider a network with three nodes w_1, w_2, and w_3 and two paths $w_1 \to w_2$ with weight G_{21} and $w_2 \to w_3$ with weight G_{32}. Lifting the path through w_2 means that the paths $w_1 \to w_2$ and $w_2 \to w_3$ are deleted and the path $w_1 \to w_3$ with weight $G_{32}G_{21}$ is added, as shown in Figure 2.

In general, this procedure can be performed in two ways. In the first approach, every single path through $w_\alpha(t)$ results in a new module and therefore this approach is referred to as single-path immersion.

For a particular node, *local inputs* are all nodes and inputs that have direct paths towards this node and *local outputs* are all nodes that have direct paths from this node.

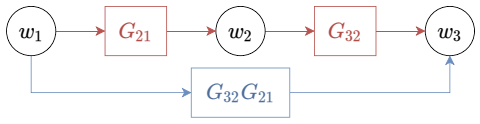


Fig. 2. The path through w_2 is lifted by deleting the red paths and adding the blue path.

Algorithm 2. (Single-path immersion). An abstraction of a module dynamic network is graphically obtained through single-path immersion by taking the following steps:

(1) Select a node $w_\alpha(t)$.
(2) Lift the paths from each local input, through $w_\alpha(t)$, to each local output.
(3) Delete the isolated $w_\alpha(t)$ from the network.
(4) Repeat the procedure to abstract more nodes.

A part of the network that is present in multiple paths in the original dynamic network appears in multiple modules arising from single-path immersion. Due to this, shared hidden states are introduced and the network structure changes.

In a second approach of lifting the paths through $w_\alpha(t)$, all paths through $w_\alpha(t)$ together result in a new module and therefore this approach is referred to as multi-path immersion.

Algorithm 3. (Multi-path immersion). An abstraction of a module dynamic network is graphically obtained through multi-path immersion by Algorithm 2 with the modification that in step (2) the paths from all local inputs, through $w_\alpha(t)$, to all local outputs are lifted together to create one (multi-variate) module.

Algorithm 3 allows for MIMO modules and therefore only one module arises during multi-path immersion and hence, no shared hidden states are introduced. The main advantage of this approach is that multi-path immersion can be seen as *zooming out* of the network and excluding some detailed structural information.

Example 2. (Single-path immersion). Consider a dynamic network with state-space description

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \\ x_3(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 0 & a_{13} \\ a_{21} & 0 & a_{23} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ b_{31} \end{bmatrix} r_1(t).$$

Its SS dynamic network is shown in Figure 3a with $G_{13} = q^{-1}a_{13}$, $G_{21} = q^{-1}a_{21}$, $G_{23} = q^{-1}a_{23}$, $R_{31} = q^{-1}b_{31}$, and with nodes $w_i = x_i$. Suppose that w_3 is abstracted from the network by single-path immersion. The abstraction is shown in Figure 3b with

$$\hat{R}_{11} = q^{-2}a_{13}b_{31}, \qquad \hat{R}_{21} = q^{-2}a_{23}b_{31}.$$

The dynamics of $R_{31} = q^{-1}b_{31}$ appears in both modules and hence, w_3 has become a shared hidden state.

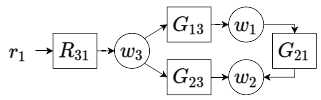
Example 3. (Multi-path immersion). Consider the SS dynamic network of Example 2 and suppose that w_3 is abstracted from the network by multi-path immersion. The abstraction is shown in Figure 3c with

$$\hat{R}_1 = \begin{pmatrix} q^{-2}a_{13}b_{31} \\ q^{-2}a_{23}b_{31} \end{pmatrix}.$$

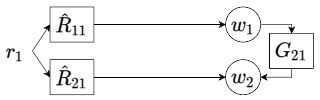
Only one module results from immersion and hence, w_3 has not become a shared hidden state.

3.2 Realisation of module dynamic networks

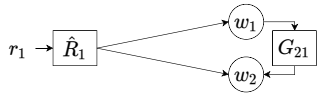
The major difference between module dynamic networks and SS dynamic networks is that in SS dynamic networks the modules are one-dimensional state-space descriptions, while in module dynamic networks the modules contain



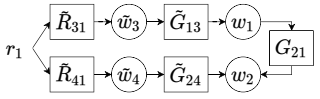
(a) The SS dynamic network of Example 2 and 3.



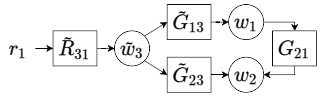
(b) The abstraction of w_3 in (a) by single-path immersion.



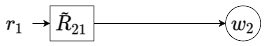
(c) The abstraction of w_3 in (a) by multi-path immersion.



(d) A realisation of the module dynamic network in (b).



(e) A realisation of the module dynamic network in (c).



(f) The abstraction of w_1 in (b), corresponding to Example 6.

Fig. 3. An SS dynamic network, three abstractions of it and realisations of two of the abstractions.

higher order dynamics which have many possible state-space realisations. A natural step to go from a general module dynamic network to an SS dynamic network is by introducing nodes, that is, by turning state variables that correspond to a particular module into node signals and thereby decreasing the order of the dynamic terms in the modules of the network. The process of transforming a general module dynamic network into an SS dynamic network is called *realisation*.

Algorithm 4. (Realisation). A realisation (SS dynamic network) of a module dynamic network is obtained by taking the following steps:

(1) Select a module $M_\alpha(q)$.
(2) Replace $M_\alpha(q)$ by a state-space realisation of it.
(3) Turn all state variables into node signals and find the new modules in accordance with Definition 4.
(4) Repeat the procedure to realise all modules.

It is well-known that realisation is a non-unique process and therefore the (number of) node signals added to the network depends on the state-space realisation. A minimum number of node signals is added if minimal state-space realisations are substituted into the modules. When using an observable canonical form, the outputs of the module are state variables in the new network. Realisation can be seen as zooming in on the network and including more detailed structural information.

Instead of finding realisations of all modules (to obtain an SS dynamic network), one can choose to find realisations of only some modules and instead of turning all state variables into node signals, one can choose to introduce only one or some nodes.

Example 4. (Realisation with shared hidden states). Consider the module dynamic network of Figure 3b with

$$\hat{R}_{11} = q^{-2}\alpha_1, \qquad \hat{R}_{21} = q^{-2}\alpha_2, \qquad G_{21} = q^{-1}\alpha_3.$$

A realisation (SS dynamic network) is found through Algorithm 4 and is shown in Figure 3d with

$$\tilde{G}_{13} = q^{-1}\beta_1,\ \tilde{G}_{24} = q^{-1}\beta_2,\ \tilde{R}_{31} = q^{-1}\beta_3,\ \tilde{R}_{41} = q^{-1}\beta_4,$$

where $\beta_1\beta_3 = \alpha_1$ and $\beta_2\beta_4 = \alpha_2$. This SS dynamic network has a different structure than the underlying SS dynamic network shown in Figure 3a, because of the shared hidden state in $\hat{R}_{11}$ and $\hat{R}_{21}$.

Example 5. (Realisation without shared hidden states). Consider the module dynamic network of Figure 3c with

$$\hat{R}_1 = \begin{pmatrix} q^{-2}\alpha_1 \\ q^{-2}\alpha_2 \end{pmatrix}, \qquad G_{21} = q^{-1}\alpha_3.$$

A realisation (SS dynamic network) is found through Algorithm 4 and is shown in Figure 3e with

$$\tilde{G}_{13} = q^{-1}\beta_1, \qquad \tilde{G}_{23} = q^{-1}\beta_2, \qquad \tilde{R}_{31} = q^{-1}\beta_3,$$

where $\beta_1\beta_3 = \alpha_1$ and $\beta_2\beta_3 = \alpha_2$. This SS dynamic network has the same structure as the underlying SS dynamic network shown in Figure 3a.

3.3 Equivalence between module and SS dynamic networks

Now it is clear how to transform SS dynamic networks into general module dynamic networks and vice versa and that they are equivalent if the node signals remain invariant.

Proposition 1. (From SS to module dynamic network). An SS dynamic network with minimal state-space dimension n can be transformed by abstraction into a general module dynamic network with generic McMillan degree $\geq n$, where equality holds if and only if the abstraction generates no shared hidden states.

Single-path immersion does not lead to shared hidden states if it is equivalent to multi-path immersion. The local network structure already reveals whether shared hidden states are generated by abstraction.

Proposition 2. (Shared hidden states).

(a) Single-path immersion of a single node leads to a shared hidden state if and only if this node has multiple local inputs or multiple local outputs.
(b) Single-path immersion of multiple nodes leads to shared hidden states if and only if at least one of the following holds:
  - The nodes jointly have multiple local inputs and at least one of the nodes has multiple local inputs.
  - The nodes jointly have multiple local outputs and at least one of the nodes has multiple local outputs.
(c) Multi-path immersion never leads to shared hidden states.

Point (b) of Proposition 2 implies that the introduction of a shared hidden state can sometimes be nullified by removing an additional node from the network.

Example 6. (Nullified shared hidden state). Consider the abstraction of Example 2 shown in Figure 3b. Suppose that w_1 is also removed from the network by immersion. Then $\hat{R}_{11}$, $\hat{R}_{21}$ and G_{21} are combined in one module, without shared hidden states, see Figure 3f.

The reverse transformation of abstraction is the realisation of a module dynamic network into an SS dynamic network. In this process, shared hidden states are not taken into account, because their existence is unknown.

Proposition 3. (From module to SS dynamic network). A module dynamic network with generic McMillan degree n can be transformed into an equivalent SS dynamic network with state-space dimension n by realisation through Algorithm 4 using minimal state-space realisations.

The resulting SS dynamic network is not unique due to the freedom in creating minimal realisations of single modules. Further, shared hidden states represent the same node signal but are realised as different node signals. Sometimes the modelling procedure prevents for shared hidden states.

4. MORE GENERAL NETWORKS

4.1 Networks with disturbances

Dynamic networks as discussed in this paper were only considered to have known input signals (2). However, the node signals of the network can also be influenced by unknown disturbance signals. Typically, these disturbance signals are modelled as realisations of stationary stochastic processes. The network is described in matrix form by

$$w(t) = Gw(t) + Rr(t) + He(t), \tag{5}$$

where $He(t)$ represents the disturbance signals with H composed of the elements $H_{jp}(q)$, $p = 1, \ldots, P \leq L$ and with $e(t)$ the vectorised version of $e_p(t)$.

From (5) it can be seen that $e(t)$ and H have the same role as $r(t)$ and R respectively and therefore $e(t)$ and H can be considered likewise. Further, H is not part of the physics: it is just a modelling choice used for describing the unknown disturbance signals. This means that the realisations and hidden states of these modules are of less interest.

4.2 Networks with general measurements

The node signals of the dynamic networks as discussed in this paper where directly measured, but this is not always possible. The measurements can also be linear combinations of node signals and excitation signals and can be subject to additional sensor noise. The L_m measurements are then written in matrix form as

$$\tilde{w}(t) = Cw(t) + Dr(t) + s(t), \tag{6}$$

where $s(t)$ is the sensor noise.

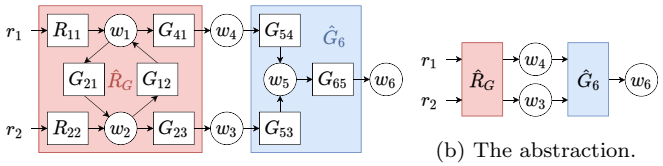
Algorithm 5. (Handling general measurements). A module dynamic network with measurements of the form (6) can be transformed into a module dynamic network with directly measured nodes by taking the following steps:

(1) Add node signals to the network that are directly measured, i.e. equal to $\tilde{w}_j(t)$, $j = 1, \cdots, L_m$.
(2) Add the corresponding modules, containing gains of the form $C(j, i)$, $D(j, k)$, and 1.

The resulting network contains unmeasured node signals $w_j(t)$, which can be removed from the network by abstraction, and measured node signals $\tilde{w}_j(t)$, which depend on the unmeasured node signals via static terms.

5. CONSTRUCTING SUB-NETWORKS

Dynamic networks often consist of sub-systems interacting with each other, where each sub-system has its own dynamics. From this point of view, a dynamic network can be seen as the interconnection of sub-networks, where a sub-network consists of several modules and at least one node. A network can be partitioned into sub-networks by drawing boxes around certain areas in the network and grouping the interior of a box into a new module. The boxes should be non-overlapping, their terminals should be connected to inputs (r) or nodes (w) and each box should include all nodes between the modules in the box.



(a) The original dynamic network.

(b) The abstraction.

Fig. 4. A module dynamic network with two sub-networks (the red and the blue box) and its abstraction.

This method is equivalent to abstracting all nodes in the box by multi-path immersion and analogous to zooming out of the network, that is, viewing the network at a higher level, where the network consists of less modules and nodes.

Example 7. (Sub-networks). Consider the module dynamic network of Figure 4a, where the red and blue box indicate sub-networks. The interiors of the red and the blue box are captured in the new modules $\hat{R}_G$ and $\hat{G}_6$ respectively, as shown in Figure 4b. This is the same as abstracting w_1 and w_2 in the red box and w_5 in the blue box by multi-path immersion.

6. DISCUSSION

Lemma 1 implies that module dynamic networks with self-loops in general are never a unique representation and thus cannot uniquely be identified from measurement data. This is not true for SS dynamic networks, because their modules have a specific structure. The identifiability problem can be solved by eliminating self-loops from the network, but this can introduce shared hidden states.

In the current literature, identification is often applied in a module dynamic network without self-loops and restricted to SISO modules. This implies that if there is a (physical) state-space form underneath the module dynamic network, shared hidden states can be present that will not be recognised as such, when all modules are independently parameterised as SISO modules. In order to appropriately deal with this situation (and arrive at minimum variance results for estimated models) the handling of MIMO parameterised modules would be necessary. From an identification perspective, this is the primal lesson to be learned from the analysis in this paper.

The network structure cannot completely be identified through its modules if a dynamic network contains hidden states. Only the network structure that manifests itself in transfer functions between inputs and node signals can be identified. The remaining structure is hidden in the modules and shared hidden states will remain undetected.

7. CONCLUSIONS

A module dynamic network has been formalised as a model for describing dynamic networks. This concept has been extended by allowing self-loops and MIMO modules. As a result, the module dynamic network incorporates state-space forms as a special case.

A module dynamic network allows to zoom in/out on/of the network to include/exclude more detailed structural information. Zooming in/out is represented by an increased/decreased number of node signals and a decreased/increased order of the module dynamics. Zooming in takes place by realisation (replacing modules by state-space realisations), which in construction is non-unique. Zooming out takes place by multi-path immersion (removing measured node signals from the network), in which the loss of structural information by the introduction of shared hidden states is avoided.

The main question in this paper was: can a state-space form always be converted into a module representation without losing any information and vice versa? The answer to this question is: yes, provided that MIMO modules are allowed in the module dynamic networks.

REFERENCES

Chang, Y.H., Gray, J.W., and Tomlin, C.J. (2014). Exact reconstruction of gene regulatory networks using compressive sensing. *BMC Bioinformatics*, 15(1).

Chetty, V. and Warnick, S. (2015). Network semantics of dynamical systems. In *2015 54th IEEE Conference on Decision and Control (CDC)*, 1557–1562.

Dankers, A., Van den Hof, P.M.J., Bombois, X., and 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., Van den Hof, P.M.J., Bombois, X., and Heuberger, P.S.C. (2016). Identification of dynamic models in complex networks with prediction error methods: Predictor input selection. *IEEE Transactions on Automatic Control*, 61(4), 937–952.

Dörfler, F. and Bullo, F. (2013). Kron reduction of graphs with applications to electrical networks. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60(1), 150–163.

Gonçalves, J., Howes, R., and Warnick, S. (2007). Dynamical structure functions for the reverse engineering of lti networks. In *2007 46th IEEE Conference on Decision and Control (CDC)*, 1516–1522.

Karcanias, N., Sagianos, E., and Milonidis, E. (2005). Structural identification: The computation of the generic mcmillan degree. In *2005 44th IEEE Conference on Decision and Control (CDC)*, 7222–7227.

Kivits, E.M.M. and Van den Hof, P.M.J. (2018). On representations of linear dynamic networks. Technical report, Eindhoven University of Technology. URL `http://publications.pvandenhof.nl/Reportfiles/Kivits&VandenHof_SYSID2018_report.pdf`.

Materassi, D. and Innocenti, G. (2010). Topological identification in networks of dynamical systems. *IEEE Transactions on Automatic Control*, 55(8), 1860–1871.

Materassi, D. and Salapaka, M.V. (2015). Identification of network components in presence of unobserved nodes. In *2015 54th IEEE Conference on Decision and Control (CDC)*, 1563–1568.

Risuleo, R.S., Bottegal, G., and Hjalmarsson, H. (2017). Variational bayes identification of acyclic dynamic networks. *IFAC-PapersOnLine*, 50(1), 10556–10561. 20th IFAC World Congress.

Van den Hof, P.M.J., Dankers, A., Heuberger, P.S.C., and 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.

Warnick, S. (2015). Shared hidden state and network representations of interconnected dynamical systems. In *2015 53rd Allerton Conference on Communication, Control, and Computing*, 25–32.

Weerts, H.H.M. and Linder, J. (2018). Personal communication.

Weerts, H.H.M., Van den Hof, P.M.J., and Dankers, A.G. (2018). Identifiability of linear dynamic networks. *Automatica*, 89, 247 – 258.

Woodbury, N., Dankers, A., and Warnick, S. (2017). On the well-posedness of lti networks. In *2017 56th IEEE Conference on Decision and Control (CDC)*, 4813–4818.

Yeung, E., Gonçalves, J., Sandberg, H., and Warnick, S. (2010). Representing structure in linear interconnected dynamical systems. In *2010 49th IEEE Conference on Decision and Control (CDC)*, 6010–6015.

Yeung, E., Gonçalves, J., Sandberg, H., and Warnick, S. (2011). Mathematical relationships between representations of structure in linear interconnected dynamical systems. In *2011 American Control Conference (ACC)*, 4348–4353.