Version 22-12-2017. Submitted for presentation at 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 consequences 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 representation allows to zoom in/out on the network to include/exclude more detailed structural information. Zooming in and out on the network is respectively described by realisation and 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. Mason (1953), Mason (1956), Lunze (1992), Wasserman and Faust (1994), Mantegna (1999), Urban and Keitty (2001), Sontag et al. (2004), Materassi and Innocenti (2009), and Paré et al. (2013).

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, see also Boccaletti et al. (2006), Mesbahi and Egerstedt (2010), and Cheng et al. (2016). The weights can also be dynamic transfer functions, which relates closely to the dynamical structure function described by Gonçalves et al. (2007) and the module representation of Van den Hof et al. (2013).

A dynamic network formulation in an identification context has been introduced recently 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 between (measured) node signals. These transfer function links are referred to as modules. The dynamic network formulation starts with a module representation, but is re-written into a dynamical structure function, which is later related to the overall transfer function of the network in Weerts et al. (2015) and Weerts et al. (2018). The missing part is the relation to the state-space description.

In terms of identification in dynamic networks, the following problems have been addressed so far: the identification of a single module: Van den Hof et al. (2013), Gevers and Bazanella (2015), Materassi and Salapaka (2015), Dankers et al. (2015), Everitt et al. (2016), Dankers et al. (2016); the identification of all modules: Weerts et al. (2017), Risuleo et al. (2017); the identification of the structure or topology: Materassi and Innocenti (2009), Materassi and Innocenti (2010), Hayden et al. (2014); and the identifiability of the network: Gonçalves and Warnick (2008), Weerts et al. (2015), Bazanella et al. (2017), Weerts et al. (2018).

Some different representations of dynamic networks are presented in Yeung et al. (2010), Yeung et al. (2011), Chetty and Warnick (2015), and Warnick (2015). These 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.

⋆ Concept paper in preparation for submission to the 19th IFAC Symposium on System Identification, July 2018, Stockholm, Sweden. Version 22 December 2017. 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).

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 contains the extension to general measurements. Section 5 describes the division of a network 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 measured variables or nodes $w_j(t)$, $j = 1, \dots, L$, and K known signals $r_k(t)$, $k = 1, \dots, K$. Each node signal is equal to

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

where $G_{ji}(q)$ and $R_{jk}(q)$ are proper rational transfer functions, $r_k(t)$ is a quasi-stationary external excitation signal, $v_j(t)$ is an unmeasured disturbance signal being a realisation of a stationary stochastic process, and q^{-1} is 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 $w_j(t)$, $w_i(t)$, $r_k(t)$, and $v_j(t)$ 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.

The measured node signals can be subject to additional sensor noise. Since the measurements are not in the centre of attention, the sensor noise is omitted for simplicity.

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

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

with matrices G and R composed of elements $G_{ji}(q)$ and $R_{jk}(q)$ respectively, and where $w(t)$, $r(t)$, and $v(t)$ are vectorised versions of $w_j(t)$, $r_k(t)$, and $v_j(t)$ 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 of the form $y = Qy + Pu$, as introduced by Gonçalves et al. (2007), with $Q = G$, $P = [R\ I]$, $y = w$, and $u = \begin{bmatrix} r^\top & v^\top \end{bmatrix}^\top$.

Figure 1a shows a single building block of a module dynamic network as described in (1).

Definition 1. (Module dynamic network). A module dynamic network is a network consisting of interconnections of nodes and excitations through transfer function matrices (or modules), as defined in (1). If all signals are scalar-valued, this is referred to as a SISO module dynamic network.

A particular property of the module dynamic networks in literature is that they only allow for SISO modules and that they exclude self-loops. This choice has been made in order 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.

Equivalent here means that the node signals $w_j(t)$ remain invariant. 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.

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 2. (Generic McMillan degree). The generic McMillan degree of a system is the McMillan degree of the system when all parameters take generic values.

Generic values are numerical values that preserve the network structure and avoid special situations, such as pole-zero cancellations. Hence, the generic McMillan degree of a network prevents for pole-zero cancellations when all modules are merged and thus also when an arbitrary number of modules is merged.

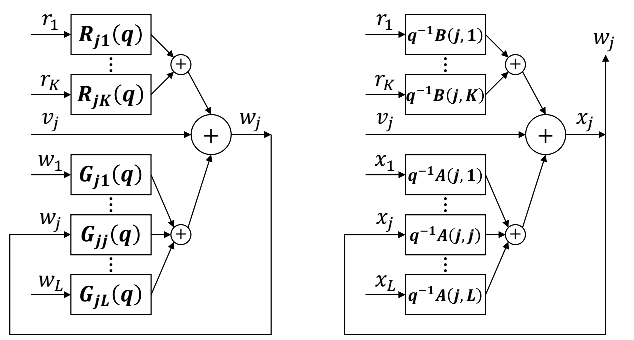
Lemma 2. (Generic McMillan degree). The generic McMillan degree of a network is equal to the sum of the generic McMillan degrees of all modules in the network.

2.2 State-space (SS) dynamic networks

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

$$x(t+1) = Ax(t) + Br(t) + v(t), \quad w(t) = x(t), \quad (3)$$

where $x(t)$ is the state variable, $r(t)$ is a quasi-stationary external excitation signal, $v(t)$ is an unmeasured disturbance signal, and $w(t)$ is the measured output variable. It is typically assumed that all state variables are directly measured.



(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 dynamic network and of an SS dynamic network.

A state-space description can also be depicted as a dynamic network. Moreover, this dynamic network is a special form of a SISO module dynamic network, where the state variables are the node signals of the network. Figure 1b shows a single building block of a state-space dynamic network as described in (3).

Definition 3. (State-space (SS) dynamic network). A state-space (SS) dynamic network is a SISO module dynamic network, as defined in Definition 1, with the additional properties that

- Every state variable $x_j(t)$ is a node signal.
- Every module between node signals has the form $G_{ji}(q) = q^{-1}A(j,i)$.
- Excitations r_k to node j have modules of 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 Immersion 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 (measured) node signals and thereby increasing the order of the dynamic terms in the modules of the network.

The process of removing unmeasured node signals from the network is called immersion and has been worked out in Dankers et al. (2016) for SISO module dynamic networks without self-loops. In immersion, the paths through nodes are lifted and isolated nodes are deleted. Lifting the path abc means that the paths ab and bc are deleted and the path ac is added. The resulting network is called an *immersed* dynamic network, in which the nodes behave exactly the same as in the original network and the remaining node signals stay invariant. The immersion process of Dankers et al. (2016) is generalised to dynamic networks with self-loops in the following algorithm.

Algorithm 1. (Single-path immersion). An immersed dynamic network is obtained from a dynamic network by taking the following steps:

(1) Select an unmeasured node.
(2) Find all nodes (w) and inputs (r, v) connected to this unmeasured node. The nodes and inputs with paths to the unmeasured node are denoted as *local inputs*. The nodes and inputs with paths from the unmeasured node are denoted as *local outputs*.
(3) Lift the paths from each local input, through the unmeasured node, to each local output.
(4) Delete the unmeasured node from the network.
(5) If there is an unmeasured node left go to step 1, otherwise quit.

Unmeasured state variables are removed from the network in the sense that they are not node signals anymore. These states 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).

Algorithm 2. (Multi-path immersion). An immersed dynamic network can also be obtained from a dynamic network by Algorithm 1 with the modification that in step (3) the paths from all local inputs, through the unmeasured node, to all local outputs are combined to create one (multi-variate) module.

Algorithm 2 allows for MIMO modules and thus less modules arise during immersion. The main disadvantage of these modules is that a part of the network structure disappears into the modules. Note that single-path and multi-path immersion are equivalent for a node (or a group of nodes) with only one local input and one local output. Immersion can be seen as zooming out on the network and excluding some detailed structural information.

Example 1. (Single-path immersion). Consider a dynamic network with state-space description

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

where all states are measured. Its SS dynamic network is shown in Figure 2a with

$G_{13} = q^{-1}a_{13}$, $G_{21} = q^{-1}a_{21}$, $G_{23} = q^{-1}a_{23}$, $R_{31} = q^{-1}b_{31}$.

Suppose that $x_3(t)$ is not measured anymore and that it is removed from the network by single-path immersion. The immersed dynamic network is shown in Figure 2b with

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

Since x_3 has become a shared hidden state, the dynamics of R_{31} appears in both modules.

Example 2. (Multi-path immersion). Consider the SS dynamic network of Example 1 and suppose that $x_3(t)$ is removed from the network by multi-path immersion. The immersed dynamic network is shown in Figure 2c with

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

Since only one module results from immersion, $x_3(t)$ 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 higher order dynamics which have many possible state-space realisations. The process of transforming a general module dynamic network into an SS dynamic network is called realisation.

Algorithm 3. (Realisation). A state-space realisation (SS dynamic network) of a module dynamic network is obtained by taking the following steps:

(1) Select a module.
(2) Replace the module by a state-space realisation of it.

(3) Turn all state variables into node signals and find the modules between them.
(4) If there is any module without a state-space realisation left go to step 1, otherwise quit.

Step (2) does not change the behaviour of any module or node signal in the network. Step (3) adds node signals to the network, but the already existing node signals remain invariant. A minimum number of node signals is added if minimal state-space realisations are substituted into the modules. Realisation can be seen as zooming in on the network and including more detailed structural information.

Example 3. (Realisation with shared hidden states). Consider the SISO module dynamic network of Figure 2b with

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

A state-space realisation (SS dynamic network) is found through Algorithm 3 and is shown in Figure 2d 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 2a, caused by the shared hidden state in $\hat{R}_{11}$ and $\hat{R}_{21}$.

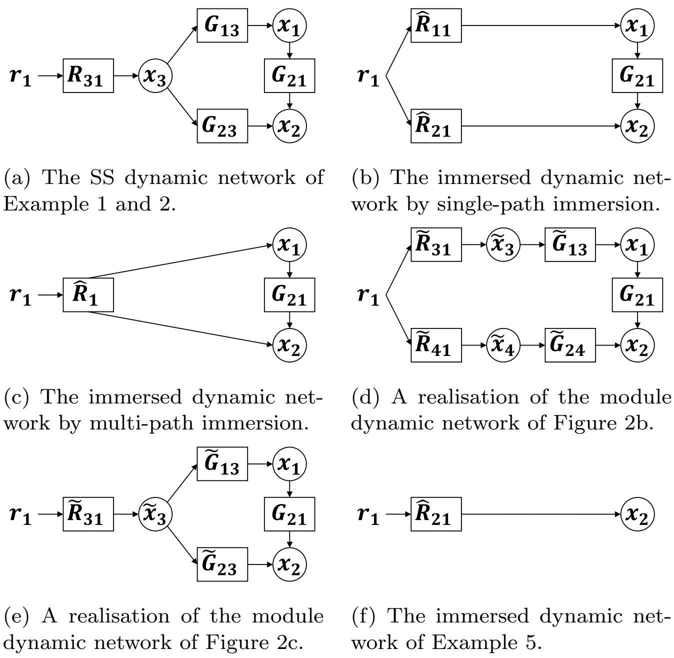
Example 4. (Realisation without shared hidden states). Consider the module dynamic network of Figure 2c with

$$\hat{R}_1 = \begin{pmatrix} q^{-2}\alpha_1 \\ q^{-2}\alpha_2 \end{pmatrix}.$$

A state-space realisation (SS dynamic network) is found through Algorithm 3 and is shown in Figure 2e 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 2a.



(a) The SS dynamic network of Example 1 and 2.

(b) The immersed dynamic network by single-path immersion.

(c) The immersed dynamic network by multi-path immersion.

(d) A realisation of the module dynamic network of Figure 2b.

(e) A realisation of the module dynamic network of Figure 2c.

(f) The immersed dynamic network of Example 5.

Fig. 2. An SS dynamic network, three immersed dynamic networks of it, and realisations of two of the immersed dynamic networks.

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 immersion into a general module dynamic network with generic McMillan degree $\geq n$, where equality holds if and only if the immersed dynamic network has no shared hidden states.

Single-path immersion does not lead to shared hidden states when it is equivalent to multi-path immersion. The local network structure already shows whether shared hidden states are introduced by immersion.

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.

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

Example 5. (Nullified shared hidden state). Consider the immersed dynamic network of Example 1, see Figure 2b. Suppose that node x_1 is also removed from the network. Then $\hat{R}_{11}$, $\hat{R}_{21}$, and G_{21} are merged into one module, without shared hidden states, see Figure 2f.

The reverse transformation of immersion 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 3.

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. To wit, when each module represents a distinct part of the network.

4. NETWORKS WITH GENERAL MEASUREMENTS

Dynamic networks as discussed in this paper were only considered to have measurements equal to the state variables of the network (3). It is not always possible to measure the states. The measurements can also be linear combinations of states and excitation signals and can be subject to additional sensor noise. The L_m measurements are then written in matrix form as

$$w(t) = Cx(t) + Dr(t) + s(t), \tag{4}$$

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

Algorithm 4. (From general to node measurements). A module dynamic network with measurements of the form (4) can be transformed into a module dynamic network with measured node signals by taking the following steps:

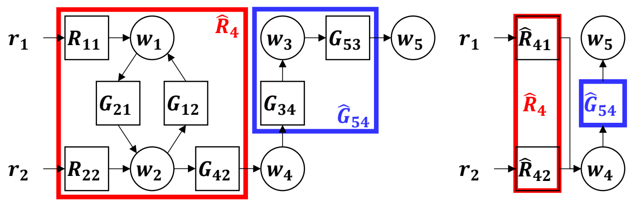
(1) Add node signals to the network that are directly measured, i.e. equal to $w_j(t)$, $j = 1, \cdots, L_m$.
(2) Add the corresponding modules, containing gains of the form $C(j,i)$, $D(j,k)$, and 1.

The resulting network contains unmeasured node signals $x_j(t)$, which can be removed from the network by immersion, and measured node signals $w_j(t)$, 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. 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 and such that its terminals are connected to inputs (r, v) or nodes (w). This method is equivalent to removing all node signals in the box from the network by immersion and is analogues to zooming out on the network.

Example 6. (Sub-networks). Consider the module dynamic network of Figure 3a, where the red and blue box indicate sub-networks. The interior of the blue box can be captured in a single module, as shown in Figure 3b. This is the same as removing node w_3 from the network by (either form of) immersion. Capturing the interior of the red box in a single module is the same as removing node w_1 and w_2 from the network by multi-path immersion. This results in a module with two inputs and one output ($\hat{R}_4$ in Figure 3b). Single-path immersion results in two SISO modules ($\hat{R}_{41}$ and $\hat{R}_{42}$ in Figure 3b) with shared hidden states.



(a) A module dynamic network with two sub-networks (red and blue box).

(b) The immersed dynamic network.

Fig. 3. A module dynamic network with two sub-networks and the corresponding immersed dynamic network.

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 in an identification setting, 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 in identification would be necessary.

The network structure cannot be identified completely 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 appearing in multiple modules 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 of SISO module dynamic networks. A module dynamic network allows to zoom in/out on 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 immersion (removing measured node signals from the network), of which two forms are discussed: single-path and multi-path immersion. Single-path immersion can cause loss of information in the model transforms due to the occurrence of shared hidden states, which are not recognised in the module representation. The loss of structural information by the introduction of shared hidden states is avoided in multi-path immersion.

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

Bazanella, A.S., Gevers, M., Hendrickx, J.M., and Parraga, A. (2017). Identifiability of dynamical networks: which nodes need be measured? In *2017 56th IEEE Conference on Decision and Control (CDC)*, 5870–5875.

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., and Hwang, D.U. (2006). Complex networks: Structure and dynamics. *Physics Reports*, 424(4), 175–308.

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

Cheng, X., Kawano, Y., and Scherpen, J.M.A. (2016). Graph structure-preserving model reduction of linear network systems. In *2016 European Control Conference (ECC)*, 1970–1975.

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.

Everitt, N., Bottegal, G., Rojas, C.R., and Hjalmarsson, H. (2016). Identification of modules in dynamic networks: An empirical bayes approach. In *2016 55th IEEE Conference on Decision and Control (CDC)*, 4612–4617.

Gevers, M. and Bazanella, A.S. (2015). Identification in dynamic networks: Identifiability and experiment design issues. In *2015 54th IEEE Conference on Decision and Control (CDC)*, 4005–4010.

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.

Gonçalves, J. and Warnick, S. (2008). Necessary and sufficient conditions for dynamical structure reconstruction of lti networks. *IEEE Transactions on Automatic Control*, 53(7), 1670–1674.

Hayden, D., Yuan, Y., and Gonçalves, J. (2014). Network reconstruction from intrinsic noise: Minimum-phase systems. In *2014 American Control Conference (ACC)*, 4391–4396.

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`.

Lunze, J. (1992). *Feedback Control of Large-Scale Systems*. Series in Systems and Control Engineering. Prentice Hall International.

Mantegna, R.N. (1999). Hierarchical structure in financial markets. *The European Physical Journal B - Condensed Matter and Complex Systems*, 11(1), 193–197.

Mason, S.J. (1953). Feedback theory - some properties of signal flow graphs. *Proceedings of the IRE*, 41(9), 1144–1156.

Mason, S.J. (1956). Feedback theory - further properties of signal flow graphs. *Proceedings of the IRE*, 44(7), 920–926.

Materassi, D. and Innocenti, G. (2009). Unveiling the connectivity structure of financial networks via high-frequency analysis. *Physica A: Statistical Mechanics and its Applications*, 388(18), 3866–3878.

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.

Mesbahi, M. and Egerstedt, M. (2010). *Graph theoretic methods in multiagent networks*. Princeton University Press.

Paré, P.E., Chetty, V., and Warnick, S. (2013). On the necessity of full-state measurement for state-space network reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, 803–806.

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.

Sontag, E., Kiyatkin, A., and Kholodenko, B.N. (2004). Inferring dynamic architecture of cellular networks using time series of gene expression, protein and metabolite data. *Bioinformatics*, 20(12), 1877–1886.

Urban, D. and Keitty, T. (2001). Landscape connectivity: a graph-theoretic perspective. *Ecology*, 82(5), 1205–1218.

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 Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 25–32.

Wasserman, S. and Faust, K. (1994). *Social network analysis: methods and applications*. Cambridge University Press.

Weerts, H.H.M., Dankers, A.G., and Van den Hof, P.M.J. (2015). Identifiability in dynamic network identification. *IFAC-PapersOnLine*, 48(28), 1409–1414. 17th IFAC Symposium on System Identification SYSID 2015.

Weerts, H.H.M., Van den Hof, P.M.J., and Dankers, A.G. (2017). Identification of dynamic networks with rank-reduced process noise. *IFAC-PapersOnLine*, 50(1), 10562–10567. 20th IFAC World Congress.

Weerts, H.H.M., Van den Hof, P.M.J., and Dankers, A.G. (2018). Identifiability of linear dynamic networks. *Automatica*. To appear. arXiv preprint arXiv:1609.00864.

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.