Characterizing Network Controllability and Observability for Abstractions and Realizations of Dynamic Networks

Charles A. Johnson * **Sean Warnick** *

* *Information and Descision Algorithms Laboratories, Computer Science Department, Brigham Young University, Provo, UT 84602 USA (e-mail: charles.addisonj@byu.edu, sean@cs.byu.edu).*

Abstract: One method for managing the complexity of a dynamic network is to abstract some of this complexity away by using a simpler, yet behaviorally equivalent, mathematical model. A theory for such abstractions is currently under development (cf. Kivits and Van den Hof (2018) as well as Woodbury and Warnick (2019)). While recent work has considered concepts of controllability and observability for networked dynamic systems (cf. Xiang et al. (2019) and Liu and Barabási (2016)), this paper analyzes these concepts for *abstractions* of dynamic networks. In particular, we present the notion of a *complete* abstraction and an *extraneous* realization of a dynamic network and show that these concepts characterize the controllability and observability properties of a class of abstractions of dynamic networks.

Keywords: Structural Controllability, Model Reduction, Dynamic Networks

1. INTRODUCTION

Dynamic networks are a useful representation of dynamic systems. They not only reflect the behavior of the system, but also capture a notion of structure in the way the system computes its behavior. In particular, nodes in a dynamic network represent variables, or signals, while edges between nodes describe dynamic relationships between signals. In linear time invariant (LTI) systems, these edges are labeled with SISO transfer functions (or their time-domain equivalent, a convolution kernal) characterizing the dependency of one variable on another. When considering directed edges, we align the direction of an edge with the causal dependency among variables.

A number of interesting systems questions can be posed in terms of dynamic networks. For example, work on characterizing informativity conditions for discovering network structure and edge dynamics under various conditions has been analyzed in Gonçalves and Warnick (2008), Adebayo et al. (2012), Chetty and Warnick (2017), Chetty et al. (2013) and Paré et al. (2013). Likewise, a rich body of literature is emerging on the identification of part or all of a dynamic network, see Materassi and Innocenti (2010), Materassi (2011), Quinn et al. (2015), Weerts et al. (2016) and Van den Hof et al. (2013). Distributed control problems have been explored where the dynamic network of the controller is constrained to have a particular network structure (see Rai and Warnick (2013)), and the security of cyber-physical-human systems has been explored in terms of the underlying dynamic network characterizing the system in Rai et al. (2012), Chetty et al. (2014) and Grimsman et al. (2016).

Another set of interesting questions deals with the relationship between the graph structure of a dynamic network and its behavior. Some of these questions include structural controllability and reachability (first introduced in Lin (1974)), which extend the classic notions of controllability and observability of state-space models. Structural controllability analysis may be applied to all the fields where controllability is relevant and is especially of interest as exact knowledge of the model parameters are not required Xiang et al. (2019).

However, the complexity and scale of modern-day networks means that such analysis can be prohibitively expensive. Several strategies for managing this complexity arise from considering *abstractions* (behaviorally equivalent, though structurally simplified models) of dynamic networks. A rich theory of such abstractions is currently under development, see Johnson and Warnick (2020), Woodbury and Warnick (2019) and Kivits and Van den Hof (2018).

This paper highlights key similarities and differences between dynamic networks and *networked dynamic systems*, a prominent model class for considering structured complex networks. It then contributes definitions and characterizations of network controllability and observability of abstractions of linear dynamic networks. These results highlight the advantages of considering network abstractions when modeling, as well as conditions (we namely consider an abstraction condition called *completeness*) on which doing so fundamentally changes the interpretation of the abstracted (simplified) model.

2. BACKGROUND

The terms *networked dynamic systems* and *dynamic networks* are used in the literature to mean very different things. Likewise, different notions of controllability and observability have also been developed for networks. This section briefly surveys these concepts and establishes the focus of the results presented in this paper.

2.1 Networked Dynamic Systems vs. Dynamic Networks

Networked dynamic systems are interconnections of subsystems that result in a larger, more complicated system (Chetty and Warnick (2017); van Waarde et al. (2019)). Often such models are used to describe multi-vehicle systems (Murray (2007)), power systems (Hill and Chen (2006)), or other systems interconnected by communication networks (Neely (2010)). The typical linear model for such a system would consider linear state equations for each agent,

$$\begin{aligned} \dot{x}_i(t) &= A_i x_i(t) + B_i u_i(t) \\ y_i(t) &= C_i x_i(t) \end{aligned}$$

where $i \in 1, 2, ..., N$ indicates the agent index (in a system with N agents); $x_i(t) \in \mathbb{R}^{n_i}$ is the agent's internal state vector; the agent's input vector, $u_i(t) \in \mathbb{R}^{m_i}$, receives input signals from either external sources or neighbors in an interconnection (possibly directed) graph, $\mathcal{G}$, characterizing the structure of the entire system; the agent's output vector, $y_i(t) \in \mathbb{R}^{p_i}$, sends information to p_i neighbors defined by the same interconnection graph, $\mathcal{G}$; and t may be either continuous or discrete (in the discrete case, $\dot{x}(t)$ should be interpreted as $x(t+1)$ instead of $\frac{dx}{dt}$). Note that stacking the equations for each agent leads to a linear state space model characterizing the entire system, although the structure of $\mathcal{G}$ might get lost in such a representation. In such networks, *nodes* of $\mathcal{G}$ represent *systems* while *edges* carry *signals* transmitting information from one subsystem to another. We often call $\mathcal{G}$ the *subsystem structure* of the network, since it represents the interconnection structure of subsystems.

Dynamic networks, on the other hand, are representations of dynamic systems that also appeal to a graph, $\mathcal{G}$, to characterize their structure, and they have been used to model biochemical reaction networks (Yeung et al. (2015)), financial networks (Materassi and Innocenti (2009)), the attack surface for the security of cyber-physical-human systems (Grimsman et al. (2016) and Chetty et al. (2014)), and to address different network reconstruction and identification problems, see Weerts et al. (2018), Talukdar et al. (2017), Materassi and Salapaka (2015) and Gonçalves and Warnick (2008). Nevertheless, in these graphs, in contrast to networked dynamic systems, *nodes* represent *signals* while *edges* represent *systems*. Such systems are characterized by equations of the form

$$y = Wy + Vu$$

where y is a vector of length p and represents manifest measurements from the system and u is a vector of length m representing stochastic or deterministic inputs to the system. Note that, similar to networked dynamic systems, these equations may be considered in either the time or frequency domain and over discrete or continuous time. Thus, for example, to model a continuous time system in the frequency domain, entries of y, u, W and V would be real rational functions of the Laplace variable, $s \in \mathbb{C}$, while modeling a discrete time system in the time domain suggests that entries of these variables are functions of (discrete) time, $t \in \mathbb{Z}$, and multiplication would become the convolution operation. The graph structure, in this case, however, is revealed by the adjacency structure of W and V, and entries of these matrices reveal the dynamics associated with the system, called a *module* (see Weerts et al. (2015)), represented by each edge of the graph. Because we restrict our attention to *causal modules* and *well-posed representations* (see Woodbury et al. (2017)), the resulting graph $\mathcal{G}$ can be interpreted as revealing the *causal dependencies* among the signals y and u; thus we call $\mathcal{G}$ the *signal structure* of the system.

Every causal, linear time invariant system has both a subsystem structure and a signal structure, but in general these structures are different, see Yeung et al. (2010). One example of this difference is due to the possibility of a *shared hidden state* between modules, while subsystems do not share states. The existence of shared hidden state implies that distinct modules on the dynamic network may not be separated in the dynamic system from which the dynamic network was computed; this may complicate the interpretation of a dynamical network as a physical system.

2.2 Structural Controllability and Observability

Consider an LTI state-space model:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx. \end{aligned} \tag{1}$$

We call this model (A, B, C) for convenience.

We restate the classic definition of structural controllability given by Lin (1974) via the notion of *structural zeros* [1] explained in Johnson and Warnick (2020).

To consider the structural controllability of (A, B, C) from a graph-theoretic standpoint we represent the system with a weighted digraph. We do so by considering the states, x, and the inputs, u, to be graph nodes and then characterize the system graph as the graph composition of a bipartite digraph with weighted adjacency matrix W_u and a second digraph with weighted adjacency matrix W_x, where

$$W_u = \begin{bmatrix} 0 & B \\ 0 & 0 \end{bmatrix} \text{ and } W_x = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix}.$$

This gives us a new graph to consider, whose weighted adjacency matrix is of the form:

$$W_{x+u} = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}.$$

The first n nodes in this resulting digraph are the state-variables, x, the remaining m nodes are the inputs, u. Each zero entry in A or B can either correspond to an edge with a zero weight or to a non-existent edge. One can think of each non-existent edge on this graph as a structural zero.

Definition 1. (Structurally Controllable). The model in Equation 1 is structurally controllable if there exists a con-

[1] Loosely speaking, structural zeros correspond to the absence of a connection between variables as opposed to the existence of a potentially zero-strength connection.

trollable system $(\bar{A}, \bar{B}, \bar{C})$ which has the same structural zeros as the system (A, B, C). □

Graph structure conditions on when a system is structurally controllable (specifically path connectivity from input nodes and the absence of graph dilations) are then given for the graph structure in Lin (1974).

For these LTI systems, the same graph results on structural observabillity may be defined on a sort of graphic dual of W_{x+u}. By graphic dual we mean the $n+p$ node graph (where nodes are the states and outputs) with the weighted adjacency matrix of the form:

$$W_{x+y} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix}.$$

Structural observability can then be posed as a structural controllability problem on the transpose digraph given by

$$W_{x+y}^T = \begin{bmatrix} A^T & C^T \\ 0 & 0 \end{bmatrix}.$$

Thus we extend our definition of structural controllability to cover structural observability.

Definition 2. (Structurally Observable). The model in Equation 1 is structurally observable if there exists a controllable system $(\bar{A}^T, \bar{C}^T, \bar{B}^T)$ which has the same structural zeros as the system (A^T, C^T, B^T). □

2.3 The Computational Dynamic Network Function

The *computational dynamic network function* for LTI systems is a model derived from Equation 1, as follows. By taking the Laplace transform (or Z transform depending on if time is continuous or discrete) of Equation 1 and assuming zero initial conditions we get:

$$\begin{aligned} sX &= AX + BU \\ Y &= CX \end{aligned}$$

Now, dividing by s yields: (2)

$$\begin{aligned} X &= \frac{1}{s}AX + \frac{1}{s}BU \\ Y &= CX. \end{aligned}$$

The equation tuple $(X = \frac{1}{s}AX + \frac{1}{s}BU, Y = CX)$ is the computational dynamic network function (DNF) of (A, B, C).

We now define structural controllability and observability on the computational DNF. Since there is a one-to-one correspondence between the LTI system (A, B, C) and its computational DNF, it inherits structural controllability and observability from the LTI system it corresponds to.

Definition 3. The computational DNF, $(X = \frac{1}{s}AX + \frac{1}{s}BU, Y = CX)$, is *structurally controllable* if (A, B, I) is structurally controllable, and it is *structurally observable* if (A^T, C^T, I) is structurally controllable. □

The computational DNF may be interpreted as a digraph with the same structure as (A, B, C), but with the weights in A and B scaled by $\frac{1}{s}$, a first-order dynamic system.

2.4 General LTI Dynamic Networks

General LTI dynamic networks may be interpreted as arbitrary digraphs whose edge weights are themselves SISO

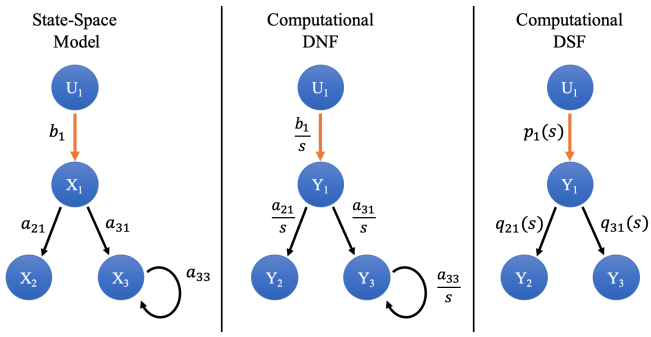

Fig. 1. Example of graphical representations of the state-space, computational DNF and computational DSF models of an LTI system. Note that there is a one-to-one correspondence between a state-space model with $C = I$ and its computational DNF, thus the two perfectly share network controllability characteristics. Note in this case that the state-space (and therefore computational DNF) model has no dilations and has path connectedness from the input to all states, so all three models (including the computational DSF) are structurally controllable.

LTI systems (typically represented by rational polynomials over $\mathbb{C}$). One such example is the computational dynamical structure function (DSF), see Gonçalves and Warnick (2008).

Example 1. Consider the state-space system:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ a_{21} & 0 & 0 \\ a_{31} & 0 & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ 0 \\ 0 \end{bmatrix} u_1$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}.$$

This system's computational DSF is the tuple $(Q(s), P(s))$, where $Y(s) = Q(s)Y(s) + P(s)U(s)$. It is computed by taking an immersion (see Definition 4) with $C_A = I$. Specifically:

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \\ Y_3(s) \end{bmatrix} = \frac{1}{s} \begin{bmatrix} 0 & 0 & 0 \\ a_{21} & 0 & 0 \\ \frac{a_{31}s}{s - a_{33}} & 0 & 0 \end{bmatrix} \begin{bmatrix} Y_1(s) \\ Y_2(s) \\ Y_3(s) \end{bmatrix} + \frac{1}{s} \begin{bmatrix} b_1 \\ 0 \\ 0 \end{bmatrix} U_1(s).$$

Note that the self-loop in the computational DNF has now been incorporated into the relationship between Y_1 and Y_3. This means that the network is structurally controllable (like the state-space model and computational DNF it was computed from), although the graph contains a dilation. □

Figure 1 demonstrates the digraphs associated with a simple LTI system, its computational DNF and its computational DSF.

3. NETWORK ABSTRACTIONS AND REALIZATIONS

A theory of abstraction has been developed for dynamic networks. This theory results in a spectrum of representations linking input-output or behavioral descriptions to state space models of the same underlying system, see Woodbury (2019) and Weerts et al. (2019). Moreover, while previous work has considered the controllability and

observability of networked dynamical systems (such as Xiang et al. (2019), Liu and Barabási (2016) and Cowan et al. (2012)), this work considers how these concepts relate to dynamic networks, in general, and their corresponding spectrum of abstractions in particular.

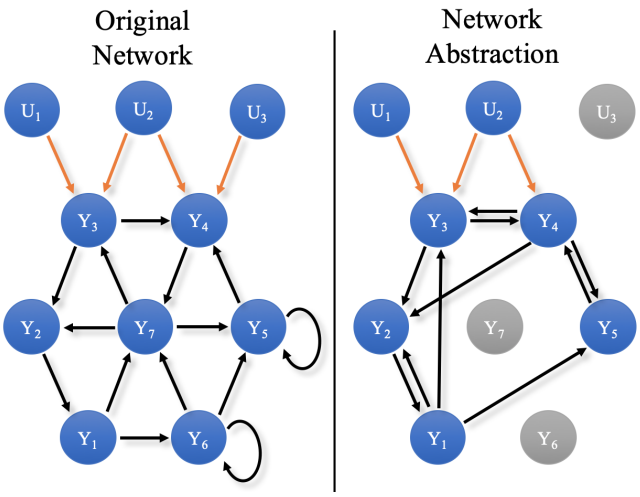

Fig. 2. A visual demonstration of the concept of a network abstraction. We apply an immersion and an input immersion. By ignoring nodes U_3, Y_6, and Y_7, we reduce the number of parameters needed to learn the simplified network model from data. Note that the edge weights (which are dynamic systems) between the two models often will not be the same.

An abstraction of a dynamic network is a behaviorally equivalent model with fewer structural parameters. While there are many forms of abstractions, we will concern ourselves with immersions in this work. Formally we define an immersion as follows (note the relationship of this definition to those given by Woodbury and Warnick (2019) and Weerts et al. (2019)).

Definition 4. (Immersion and Realization). An *immersion* of an LTI network model, $(W_R(s), V_R(s))$, with input nodes $u_R(s)$, manifest nodes $x_R(s)$ and transfer function $G_R(s)$, is a second LTI network model $(W_A(s), V_A(s))$, with input nodes $u_A(s)$, manifest nodes $x_A(s)$ and transfer function $G_A(s)$ where:

(1) W_A is hollow (has zeros on the diagonal),
(2) $x_A(s) = C_A x_R(s)$ and
(3) $G_A(s) = C_A G_R(s)$,

where C_A is a set of rows from a permutation matrix of appropriate size.

We call $(W_R(s), V_R(s))$ is the *realization* of $(W_A(s), V_A(s))$.□

Additionally, to simplify one's network models one may abstract away a network input (many things could motivate the desire to remove or ignore a network actuator). To do so we define a second form of abstraction.

Definition 5. (Input Immersion and Realization). An *input immersion* of an LTI network model, $(W_R(s), V_R(s))$, with input nodes $u_R(s)$, manifest nodes $x_R(s)$ and transfer function $G_R(s)$, is a second LTI network model $(W_A(s), V_A(s))$, with input nodes $u_A(s)$, manifest nodes $x_A(s)$ and transfer function $G_A(s)$ where:

(1) W_R is hollow (has zeros on the diagonal),
(2) $u_A(s) = C_A u_R(s)$ and
(3) $G_A(s) = G_R(s) C_A^T$,

where C_A is a set of rows from a permutation matrix of appropriate size.

We call $(W_R(s), V_R(s))$ is the *realization* of $(W_A(s), V_A(s))$.□

3.1 Structural Controllability and Observability of Dynamic Networks and their Immersions

The graphical results characterizing structural controllability and observabillity (originating with the introduction of structural control in Lin (1974)) do not extend to a reasonable definition of structural control on general dynamic networks (a parallel argument for the failure of these results to extend is given for networked dynamic systems in Cowan et al. (2012)).

Indeed, when we define structural control to be the existence of edge dynamics that permit a structurally controllable realization, we find that path connectedness is the only notion required for structural control. However, LTI systems that are not structurally controllable may be represented (as abstractions) by Dynamic Networks that are path connected.

To see that past graph results on structural controllability do not directly carry over to LTI dynamic networks, examine the dynamic network given by the DSF in Example 1 and shown in Figure 1. When one examines the graph topology of the DSF there appears to be a graph dilation, which implies that structural controllability of the network is impossible, a conclusion shown in Lin (1974). However, when one examines the computational DNF, or the state-space model which can be realized from the same network, there is no dilation (due to the existence of a self-loop on X_3), and the system is therefore structurally controllable.

Alternatively, we can restrict our definition to constrain the choice of edge dynamics with the same graph structure and then consider if a realization of the current dynamics is itself structurally controllable. But, even this working definition has its issues. It still allows a network that was derived from a structurally uncontrollable state-space model to be realized as structurally controllable. See Example 2 and Figure 3. Further restriction on which realizations we may consider for structural controllability is necessary.

Example 2. (*Uncontrollable System, Controllable Realizations*) Consider the structurally uncontrollable state-space model

$$(A, b) = \left(\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right).$$

Taking the trivial immersion of the system results in,

$$(Q_A(s), P_A(s)) = \left(\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \frac{1}{s}\begin{bmatrix} 1 \\ 1 \end{bmatrix}\right).$$

The transfer function of this immersion is: $G(s) = P(s)$. However, we may realize this system as an abstraction of the system

$$(A, B) = \left(\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}\right),$$

which is clearly controllable (B is full row rank).

Note that the transfer function of this controllable realization,

$$G_{CR}(s) = \frac{1}{s}\begin{bmatrix}1 & 1\\ 1 & 0\end{bmatrix},$$

has a higher Smith McMillan degree than that of the original model with transfer function $G_A(s) = P_A(s)$. This difference in Smith McMillan degree motivates the below definition of structural controllability of a dynamic network. □

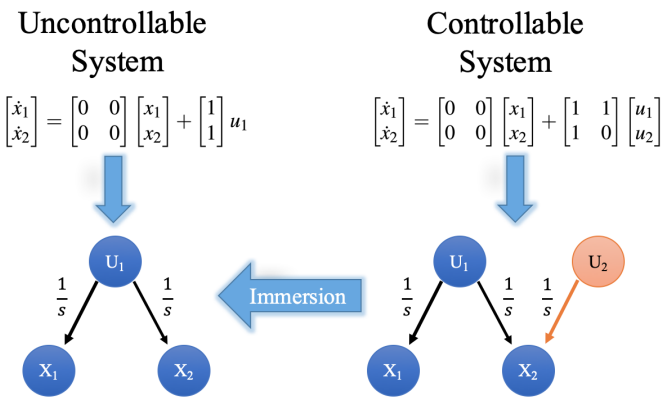

Fig. 3. In this example we see that one network can be derived from a structurally uncontrollable state-space model, but also be the abstraction of a network that was derived from a structurally controllable state-space model (hence the same network is also derivable from a structurally controllable state-space model). We thereby justify holding the Smith McMillan Degree constant when considering state-space realizations of our abstractions.

A definition of structural controllability for dynamic networks needs avoid situations like that demonstrated in Example 2. To avoid this issue we propose restricting the set of admissible realizations to those which share the same Smith McMillan degree. This will allow the realization of a class of systems which captures the same dynamic information as our network in a clear, tractable way.

Definition 6. (Structural Controllability). An LTI dynamic network, $(W_A(s), V_A(s))$, is structurally controllable if there exists a structurally controllable computational dynamic network function in the set of all possible realizations of $(W_R(s), V_R(s))$ which are of the same Smith-McMillan degree of $(W_A(s), V_A(s))$. □

Under any of the definitions we considered above (including the one we chose) we cannot necessarily trust all the basic graph conditions (see Figure 1) used to characterize structural controllability on general LTI dynamic networks because the model we reference may be derived from an equivalent model with a more complex graph structure. However, we can ask the question of when properties like structural controllability carry over from realization to immersion. As a prelude to some results that answer this question we introduce the notion of a *complete immersion* and an *extraneous realization*.

4. COMPLETE IMMERSIONS

We argue that not all network immersions are equal in their effect on the resulting model. There are some immersions which require no more than the removal of rows and columns from $W_R(s)$ and rows from $V_R(s)$, and there are some input immersions that require no more than removal of some columns of $V_R(s)$. We will refer to these as *incomplete immersions*. Incomplete immersions dramatically obfuscate the structural controllability and observability of dynamic networks by removing both structural and dynamic information from the resulting model.

Complete immersions, on the other hand, preserve underlying dynamics and, while they may appear to create dilations in the network, loop dynamics are encoded on to the edges and path connectivity is never disrupted.

Definition 7. (Complete Immersion). An (input) immersion, $(W_A(s), V_A(s))$ is complete with respect to a subset of nodes in $w_R(s)$, $w_R^*(s)$, and a subset of nodes in $u_R(s)$, $u_R^*(s)$, if $w_A(s)$ contains all the nodes $w_R^*(s)$ and $u_A(s)$ contains all the nodes of $u_R^*(s)$. □

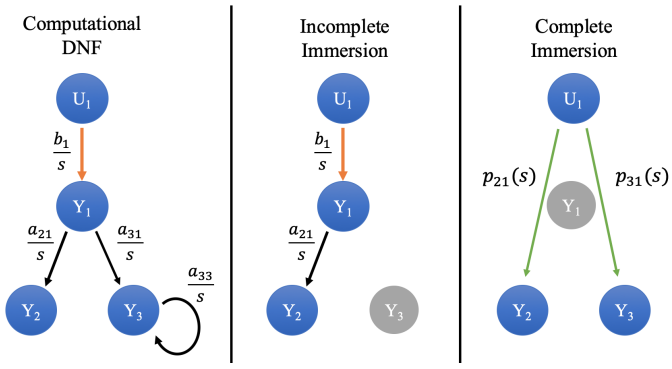

Fig. 4. An example of an incomplete and a complete immersion with respect to the set of sink nodes of the same computational DNF. When performing an immersion we throw out the abstracted nodes (hence the abstracted node is represented in gray). Note that the complete immersion required the creation of new edges. The dynamics corresponding to the edges leading in to and out of Y_1 are placed on the edges $p_{21}(s)$ and $p_{31}(s)$ to maintain equivalent transfer functions from the realization to the immersion. The preservation of these dynamics on these new edges keeps key information in the simplified model.

Example 3. (Complete and Incomplete Immersion). Consider the state-space model given in Example 1. We perform two distinct immersions on that state-space model. The first with

$$C_A = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix} \text{ and the second with } C_A = \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}.$$

The first is an incomplete immersion with respect to the set of sink nodes. It results in the DSF $(Q_I(s), P_I(s))$, where

$$Q_I(s) = \frac{1}{s}\begin{bmatrix}0 & 0\\ a_{21} & 0\end{bmatrix}, P_I(s) = \frac{1}{s}\begin{bmatrix}b_1\\ 0\end{bmatrix}.$$

Note that Q_1 and P_1 are each submatricies of the Q and P associated with the computational DSF of this state-space system. This implies that no information from the edges into and from Y_3 have been included in this immersion.

The second is a complete immersion with respect to the set of sink nodes. It results in the DSF $(Q_C(s), P_C(s))$, where

$$Q_C(s) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, P_C(s) = \frac{1}{s^2} \begin{bmatrix} b_1 a_{21} \\ \frac{b_1 a_{31}}{s - a_{33}} \end{bmatrix}.$$

In this model the entries in P_C are distinct from those of P from the computational DSF. The edge weights in P_C have been modified to encode the dynamics of the entire model. □

By definition all input immersions are incomplete with respect to source nodes.

5. RESULTS

Below we present preliminary results on the structural controllability of immersions and realizations of LTI dynamic networks.

Result 1. (Edge Weights in Incomplete Immersions). Any single-node immersion, $(W_A(s), V_A(s))$, of an LTI dynamic network, $(W_R(s), V_R(s))$, that is incomplete with respect to sink nodes where $C_A w_R(s) = w_A(s)$ may always satisfy $W_A(s) = C_A W_R(s) C_A^T$.

Proof Without loss of generality assume that we abstract away the last node and it is a sink node. So $W_R(s)$ has the block structure: $W_R(s) = \begin{bmatrix} W_{11} & 0 \\ W_{12} & w_{22} \end{bmatrix}$, with $V_R(s) = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$, where W_{12} and V_2 are row vectors. So the transfer function of the realization is: $G_R(s) = \begin{bmatrix} (I - W_{11})^{-1} V_1 \\ \frac{1}{1 - w_{22}} (W_{12}(I - W_{11})^{-1} V_1 + V_2) \end{bmatrix}$, while $G_A(s) = (I - W_{11})^{-1} V_1$. This may be achieved by an abstraction of the form: $W_A(s) = W_1 1(s)$ and $V_A(s) = V_1(s)$. ■

Result 2. (Edge Weights in Incomplete Input Immersions). Any single-node input immersion, $(W_A(s), V_A(s))$, of an LTI dynamic network, $(W_R(s), V_R(s))$, where $C_A u_R(s) = u_A(s)$ may always satisfy $V_A(s) = V_R(s) C_A^T$.

Proof Without loss of generality assume that we abstract away the last input node. So $V_R(s)$ has the block structure: $V_R(s) = [V_1 \ V_2]$, where V_2 is a column vector. So the transfer function of the realization is: $G_R(s) = (I - W_R)^{-1} [V_1 \ V_2]$, while $G_A(s) = (I - W_R)^{-1} V_1$. This may be achieved by an abstraction of the form: $W_A(s) = W_R(s)$ and $V_A(s) = V_1(s)$. ■

Note that results 1 and 2 imply that in these cases the Smith McMillan Degree of the network necessarily decreases as poles in the transfer function associated with the abstracted edges will not be used to compute any other input-output relationship in the abstraction.

Result 3. (Complete Smith McMillan Degree). The Smith McMillan degree of a single-node immersion $(W_A(s), V_A(s))$, of a weakly connected LTI dynamic network, $(W_R(s), V_R(s))$ equals that of $(W_R(s), V_R(s))$ if and only if it is complete with respect to its source and sink nodes.

Proof We prove the first direction by the contrapositive. Assume that the immersion, $(W_A(s), V_A(s))$, is not complete with respect to the set of source and sink nodes. Without loss of generality assume that we abstract away the last input node. So $W_R(s)$ has the block structure: $W_R(s) = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & w_{22} \end{bmatrix}$ and $V_R(s) = [V_1 \ V_2]$, where W_{12} and V_2 is a column vector and W_{21} is a row vector. Then, we have two cases. In the first case we have abstracted away at least one source node, so $W_{21} = 0$. In the second case we have abstracted away at least one sink node, so $W_{12} = 0$. In both cases, results 1 and 2 imply that the transfer function of the abstraction is of a lower Smith McMillan degree than the original network.

Now assume that $(W_A(s), V_A(s))$ is a complete immersion with respect to source and sink nodes of $(W_R(s), V_R(s))$. Then, to preserve transfer function equivalence, all the dynamics of the abstracted edges pass over to the transfer function of the abstraction and so the Smith McMillan degree is the same. ■

Result 4. (Structural Control of Complete Immersions). Complete immersions with respect to source nodes of a structurally controllable LTI dynamic network are structurally controllable.

Proof This is true because the Smith-McMillan degree of the immersion is that of the realization and the immersion and the immersion dynamics are consistent (the transfer function entries are equal), so the realization may be constructed from the immersion. Since that realization was structurally controllable so is the immersion. ■

Analogous results for structural observability may be acquired by applying immersions to dynamic networks generated from the dual network.

6. CONCLUSIONS

We have explored the difference between networked dynamic systems and dynamic networks. We have highlighted how dynamic networks have an established theory of abstraction which allows one to greatly reduce the topological complexity of their use and of learning them from data. However, the interpretation of the abstracted model (especially immersions of the original model) may have a significant difference in interpretation due to the existence of shared hidden state. Regardless, such abstractions preserve all of the predictive power of their topologically complex realizations when they retain the property of completeness. Specifically, we defined structural controllability of dynamic networks and noted how completeness is a necessary requirement for preserving structural controllability through the abstraction process. This informed application of dynamic network abstraction theory identifies how graph-based model topology simplification methods influences the interpretation of models of networked dynamic systems.

REFERENCES

Adebayo, J., Southwick, T., Chetty, V., Yeung, E., Yuan, Y., Gonçalves, J., Grose, J., Prince, J., Stan, G.B., and Warnick, S. (2012). Dynamical structure function identifiability conditions enabling signal structure reconstruction. In *IEEE Conference on Decision and Control*. Maui.

Chetty, V., Hayden, D., Warnick, S., and Gonçalves, J. (2013). Robust signal-structure reconstruction. In *Conference on Decision and Control*. Florence, Italy.

Chetty, V. and Warnick, S. (2017). Necessary and sufficient conditions for identifiability of interconnected subsystems. In *Conference on Decision and Control*. Melbourne, Australia.

Chetty, V., Woodbury, N., Vaziripour, E., and Warnick, S. (2014). Vulnerability analysis for distributed and coordinated destabilization attacks. In *Conference on Decision and Control*. Los Angeles.

Cowan, N.J., Chastain, E.J., Vilhena, D.A., Freudenberg, J.S., and Bergstrom, C.T. (2012). Nodal dynamics, not degree distributions, determine the structural controllability of complex networks. *PloS one*, 7(6), e38398.

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

Grimsman, D., Chetty, V., Woodbury, N., Vaziripour, E., Roy, S., Zappala, D., and Warnick, S. (2016). A case study of a systematic attack design method for critical infrastructure cyber-physical systems. In *American Control Conference*. Boston, MA.

Hill, D.J. and Chen, G. (2006). Power systems as dynamic networks. In *2006 IEEE International Symposium on Circuits and Systems*, 4–pp. IEEE.

Johnson, Charles A., W.N. and Warnick, S. (2020). Graph theoretic foundations of cyclic and acyclic linear dynamic networks.

Kivits, E. and Van den Hof, P.M. (2018). On representations of linear dynamic networks. *IFAC-PapersOnLine*, 51(15), 838–843.

Lin, C.T. (1974). Structural controllability. *IEEE Transactions on Automatic Control*, 19(3), 201–208.

Liu, Y.Y. and Barabási, A.L. (2016). Control principles of complex systems. *Reviews of Modern Physics*, 88(3), 035006.

Materassi, D. and Innocenti, G. (2010). Topological identification in networks of dynamical systems. *IEEE Transactions on Automatic Control*, 55(8), 1860–1871. doi:10.1109/TAC.2010.2042347.

Materassi, D. (2011). Reconstruction of topologies for acyclic networks of dynamical systems.

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 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. IEEE.

Murray, R.M. (2007). Recent research in cooperative control of multivehicle systems. *Journal of Dynamic Systems, Measurement, and Control*, 129(5), 571–583.

Neely, M.J. (2010). Stochastic network optimization with application to communication and queueing systems. *Synthesis Lectures on Communication Networks*, 3(1), 1–211.

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: Symposium on Network Theory*. Austin.

Quinn, C.J., Kiyavash, N., and Coleman, T.P. (2015). Directed information graphs. *IEEE Transactions on Information Theory*, 61(12), 6887–6909. doi:10.1109/TIT.2015.2478440.

Rai, A., Ward, D., Roy, S., and Warnick, S. (2012). Vulnerable links and secure architectures in the stabilization of networks of controlled dynamical systems. In *American Control Conference*, 1248–1253. Montréal, Canada.

Rai, A. and Warnick, S. (2013). A technique for designing stabilizing distributed controllers with arbitrary signal structure constraints. In *European Control Conference*. Zürich, Switzerland.

Talukdar, S., Deka, D., Materassi, D., and Salapaka, M. (2017). Exact topology reconstruction of radial dynamical systems with applications to distribution system of the power grid. In *2017 American Control Conference (ACC)*, 813–818. IEEE.

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. doi://doi.org/10.1016/j.automatica.2013.07.011. URL `http://www.sciencedirect.com/science/article/pii/S0005109813003592`. ID: 271426.

van Waarde, H.J., Tesi, P., and Camlibel, M.K. (2019). Topology identification of heterogeneous networks of linear systems. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, 5513–5518. IEEE.

Weerts, H., den Hof, P.M.J.V., and Dankers, A. (2016). Identification of dynamic networks operating in the presence of algebraic loops. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, 4606–4611. doi:10.1109/CDC.2016.7798970.

Weerts, H.H., Dankers, A.G., and Van den Hof, P.M. (2015). Identifiability in dynamic network identification. *IFAC-PapersOnLine*, 48(28), 1409–1414.

Weerts, H.H., Linder, J., Enqvist, M., and Van den Hof, P.M. (2019). Abstractions of linear dynamic networks for input selection in local module identification. *arXiv preprint arXiv:1901.00348*.

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

Woodbury, N. (2019). *Representation and Reconstruction of Linear, Time-Invariant Networks*. Ph.D Dissertation, Brigham Young Univ., Provo, UT.

Woodbury, N., Dankers, A., and Warnick, S. (2017). On the well-posedness of lti networks. In *Conference on Decision and Control*. Melbourne, Australia.

Woodbury, N. and Warnick, S. (2019). Abstractions and realizations of dynamic networks. To appear at the 2019 IEEE American Control Conference.

Xiang, L., Chen, F., Ren, W., and Chen, G. (2019). Advances in network controllability. *IEEE Circuits and Systems Magazine*, 19(2), 8–32.

Yeung, E., Gonçalves, J., Sandberg, H., and Warnick, S. (2010). Representing structure in linear interconnected dynamical systems. In *Conference on Desicion and Control*. Atlanta.

Yeung, E., Kim, J., Gonçalves, J., and Murray, R.M. (2015). Global network identification from reconstructed dynamical structure subnetworks: Applications to biochemical reaction networks. In *2015 54th IEEE Conference on Decision and Control (CDC)*, 881–888. IEEE.