2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

Mathematical Relationships Between Representations of Structure in Linear Interconnected Dynamical Systems

E. Yeung[‡], J. Gonçalves[*], H. Sandberg[○], and S. Warnick[†]

Abstract— **A dynamical system can exhibit structure on multiple levels. Different system representations can capture different elements of a dynamical system's structure. We consider LTI input-output dynamical systems and present four representations of structure: complete computational structure, subsystem structure, signal structure, and input output sparsity structure. We then explore some of the mathematical relationships that relate these different representations of structure. In particular, we show that signal and subsystem structure are fundamentally different ways of representing system structure. A signal structure does not always specify a unique subsystem structure nor does subsystem structure always specify a unique signal structure. We illustrate these concepts with a numerical example.**

I. Introduction

Interconnected dynamical systems are a pervasive component in our modern world's infrastructure. Much research has been dedicated to understanding the relationships between a system's dynamics and a system's structure, see [1], [2] for example. At the same time, research in various avenues has demonstrated that the structure of a dynamical system can be represented in more than one way. For example, [3] discusses system structure in terms of interconnected subsystems. System structure in this context refers to the sharing of variables or the linking of terminals and ports [4]. At the same time, [5], [6], [7], [8] describe system structure in terms of the dependencies between manifest variables of the system. Alternatively, system structure in the context of decentralized control can refer to the location of zero entries in a transfer function matrix.

These examples demonstrate how structure in a system can be represented in different ways. The problem of representing a system's structure using four particular representations within the LTI input-output framework is addressed in [9]. In this work, the authors demonstrate by example that a single LTI input-output system can assume multiple forms depending on the choice of structural representation.

The purpose of this paper is to discuss the relationships between the four representations of structure defined and illustrated in [9]. Instead of examining how the variation of system structure (represented in a particular way) leads to different dynamical behavior, we consider the relationships between different representations of structure that yield the same system behavior. Thus, this research is complementary to that of [1], [2], [3], [4], [5] and an extension of the research initiated by [9], [5]. The scope of our analysis will be within the framework of LTI input-output systems, a subset of the class of behavioral systems described by [4].

The rest of this paper is organized as follows. In Section II, we introduce and review four ways of representing a system's structure: complete computational, subsystem, signal, and input output sparsity structure. In Section III we present the mathematical relationships that relate these four definitions of structure. Specifically, we prove the general relationship between signal and subsystem structure, showing that information contained in one structural representation is not necessarily captured by the other. We conclude with a numerical example.

II. Representations of Structure

In this section we define four representations of system structure: complete computational structure, subsystem structure, signal structure, and input output sparsity structure. Our concept of system structure is built around an LTI input-output system G mathematically described as the generalized state-space realization

$$\begin{array}{rcl} \dot{x} &=& f(x,w,u) = Ax + \hat{A}w + Bu, \\ w &=& g(x,w,u) = \bar{A}x + \tilde{A}w + \bar{B}u, \\ y &=& h(x,w,u) = Cx + \bar{C}w + Du,. \end{array} \tag{1}$$

with $u \in \mathbb{R}^m$, $x \in \mathbb{R}^n$, $w \in \mathbb{R}^l$, $y \in \mathbb{R}^p$, $A \in \mathbb{R}^{n\times n}$, $\hat{A} \in \mathbb{R}^{n\times l}$, $\bar{A} \in \mathbb{R}^{l\times n}$, $\tilde{A} \in \mathbb{R}^{l\times l}$, $B \in \mathbb{R}^{n\times m}$, $\bar{B} \in \mathbb{R}^{l\times m}$, $C \in \mathbb{R}^{p\times n}$, $\bar{C} \in \mathbb{R}^{p\times l}$, and $D \in \mathbb{R}^{p\times m}$. By $\dot{x}$ we mean dx/dt and assume $t \in \mathbb{R}_{\geq 0}$. We refer to x, w, y, and u as the state, auxiliary, output, and input variables respectively. Moreover, here we restrict our attention to linear time invariant functions f, g and h where solutions exist for $t \geq 0$. Note that this system is in the form of a differential algebraic equation, although we will only consider systems with differentiation index zero, requiring that $(I - \tilde{A})$ is invertible. Thus, (1) is always equivalent to a standard ordinary differential or difference equation of the same order [10]. Nevertheless, we distinguish between a generalized state-space realization with auxiliary variables and a state-space realization without auxiliary variables with the following definitions.

Definition 1: Given a system (1), we call the number of auxiliary variables, l, the *intricacy* of the realization. A state-space realization (A_o, B_o, C_o, D_o) obtained from (1) by eliminating the auxiliary variables w ($l = 0$), we call a *minimal intricacy realization*.

The presence of the auxiliary variables w are used to characterize the intermediate computation in the composition

For correspondence, please contact eyeung@caltech.edu or sean.warnick@gmail.com. ‡ Control and Dynamical Systems, California Institute of Technology; † Information and Decision Algorithms Laboratories, Brigham Young University, Provo, UT; ∗ Control Group, Department of Engineering, University of Cambridge, Cambridge, UK; ○ Automatic Control Laboratory, KTH School of Electrical Engineering, Stockholm, SE.

978-1-4577-0079-8/11/$26.00 ©2011 AACC

of functions. Thus, for example, we distinguish between $f(x) = x$ and $f(x) = 2(.5)x$ by computing the latter as $f(w) = 2(w)$ and $w = g(x) = .5x$. In this way, the auxiliary variables serve to identify stages in the computation of the state-space realization (1). As we introduce subsystem structure, it will be critical to use auxiliary variables to distinguish between systems with equivalent state-space dynamics, yet structurally distinct architectures. With these preliminaries in order, we are ready to define four representations of structure.

A. Complete Computational Structure

The generalized state-space realization is a mathematical representation of the actual processes a system uses to sense its environment, represent and store variables internally, and affect change externally. The complete computational structure is a graphical representation of these processes, highlighting the structural relationships between processes that drive sensing, storage, computation, and actuation. Formally, we define complete computational structure as follows.

Definition 2: Given a system G with realization (1), its *complete* or *computational structure* is a weighted directed graph $\mathscr{C}$ with vertex set $V(\mathscr{C})$, and edge set $E(\mathscr{C})$. The vertex set contains $m+n+l+p$ elements, one associated with the mechanism that produces each input, state, auxiliary, and output variable of the system, and we label the vertices accordingly. In particular, the vertex associated with the i^{th} input is labeled u_i, $1 \leq i \leq m$, the vertex associated with the j^{th} state is labeled f_j, $0 \leq j \leq n$, the vertex associated with the j^{th} auxiliary variable is labeled g_j, $0 \leq j \leq l$, and the vertex associated with the k^{th} output is labeled h_k, $1 \leq k \leq p$. The edge set contains an edge from node i to node j if the function associated with the label of node j depends on the variable produced by node i. Moreover, the edge (i, j) is then labeled (weighted) with the variable produced by node i.

We refer to the elements of the vertex set of $\mathscr{C}$ as fundamental units of computation. Thus, our definition of structure is a deliberate choice to model system structure at a certain level of abstraction: namely, the level that obscures computational components required to perform linear algebraic operations (f, g and h vertices), integrator operations (f vertices), and the (often unknown) external computation required to produce inputs for the system. We prefer to condense these computational components as vertices, take these as the fundamental units of computation, and model the interconnection structure among these units. We do this to create a 1-to-1 correspondence with the system (1) and accordingly, define structure at the level of a generalized state-space realization.

B. Subsystem Structure

Subsystem structure refers to the appropriate decomposition of a system into constituent subsystems and the interconnection structure between these subsystems. Abstractly, it is the condensation graph of the complete computational structure graph, $\mathscr{C}$, taken with respect to a particular partition of $\mathscr{C}$ that identifies subsystems in the system. The defining property of this particular partition is *admissibility*:

Definition 3: Given a system G with realization (1) and associated computational structure $\mathscr{C}$, we say a partition of $V(\mathscr{C})$ is *admissible* if every edge in $E(\mathscr{C})$ between components of the partition represents a variable that is manifest, not hidden.

Although sometimes any aggregation, or set of fundamental computational mechanisms represented by vertices in $\mathscr{C}$, may be considered a valid subsystem, in this work a subsystem has a specific meaning. In particular, the variables that interconnect subsystems must be manifest, and thus subsystems are identified by the components of admissible partitions of $V(\mathscr{C})$. We adopt this convention to 1) enable the distinction between real subsystems vs. merely arbitrary aggregations of the components of a system, and 2) ensure that the actual subsystem architecture of a particular system is adequately reflected in the system's computational structure and associated realization, thereby ensuring that said realization is complete.

Definition 4: Given a system G with realization (1) and associated computational structure $\mathscr{C}$, the system's *subsystem structure* is a condensation graph $\mathscr{S}$ of $\mathscr{C}$ with vertex set $V(\mathscr{S})$ and edge set $E(\mathscr{S})$ given by:

- $V(\mathscr{S}) = \{S_1, ..., S_q\}$ are the elements of an admissible partition of $V(\mathscr{C})$ of maximal cardinality, and
- $E(\mathscr{S})$ has an edge (S_i, S_j) if $E(\mathscr{C})$ has an edge from some component of S_i to some component of S_j.

We label the nodes of $V(\mathscr{S})$ with the transfer function of the associated subsystem, which we also denote S_i, and the edges of $E(\mathscr{S})$ with the associated variable from $E(\mathscr{C})$.

We note that the subsystem structure of a system G always exists and is unique [9]. Traditionally, subsystem structure is mathematically represented as a linear fractional transformation (LFT) $\mathscr{F}(N, S)(s)$ with a block diagonal "subsystem" component $S(s)$ and a static "interconnection" component L (see [11] for background on the LFT). Specifically, the LFT associated with $\mathscr{S}$ will have the form

$$N = \begin{bmatrix} 0 & I \\ L & K \end{bmatrix}, \quad S(s) = \begin{bmatrix} S_1 & 0 & ... \\ 0 & \ddots & 0 \\ \vdots & 0 & S_q \end{bmatrix} \tag{2}$$

where q is the number of distinct subsystems, and L and K are each binary matrices of the appropriate dimension. To explain, the L and K matrices map the vector $\begin{bmatrix} u^T y^T \end{bmatrix}^T$ to a vector π consisting of components of y and u. The vector π has dimension corresponding to the number of columns in S, and contains the subsystem inputs for each subsystem $S_i = 1, ..., q$. Since a single manifest variable (y_j in y or u_j in u) can act as a subsystem input to multiple subsystems, π can contain repeated entries. Thus, the row dimension of $\begin{bmatrix} L & K \end{bmatrix}$ depends on the particular subsystem structure being studied, but the column dimension is always $m+p$. Note that if additional output variables are present, besides the manifest variables used to interconnect subsystems, then the structure of N and S above extend naturally. In any event, N is static and L and K are binary matrices with fixed column dimension $m+p$.

C. Signal Structure

Another way to describe the structure of a system is to characterize the direct causal dependence among each of its manifest variables; we refer to this notion of system structure as signal structure. The dynamical structure function (defined in [5] and discussed in [7], [8], [6], [12], [9]) is a representation that describes the direct causal dependence among a subset of state variables; it is the mathematical analogue of signal structure.

For the scope of this work, it suffices to show how dynamical structure functions describe direct causal dependencies among manifest variables, see [9] for a derivation. We distinguish between two kinds of causal dependencies, dynamic and static, and partition the dependencies among manifest variables accordingly in the following way:

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} Q(s) \\ C_{21} \end{bmatrix} Y_1 + \begin{bmatrix} P(s)+(I-Q(s))D_1 \\ D_2 - C_{21}D_1 \end{bmatrix} U \quad (3)$$

The matrix $\begin{bmatrix} Q(s)^T & C_{21}^T \end{bmatrix}^T$ and the matrix $\begin{bmatrix} (P(s)+(I-Q(s))D_1)^T & (D_2-C_{21}D_1)^T \end{bmatrix}^T$ we refer to as $\bar{Q}$ and $\bar{P}$ respectively and call the *generalized dynamical structure function*. Note that C_{21} and $D_2 - C_{21}D_1$ describe static direct causal dependencies among manifest variables and $Q(s)$, $P+(I-Q(s))D_1$ describe dynamic direct causal dependencies. The matrices $(Q(s),P(s))$ are called the dynamical structure function of the system (1), and they characterize a dependency graph among manifest variables as indicated in Equation (3). We note a few characteristics of $(Q(s),P(s))$ that give them the interpretation of system structure, namely:

- $Q(s)$ is a square matrix of strictly proper real rational functions of the Laplace variable, s, with zeros on the diagonal. Thus, if each entry of y_1 were the node of a graph, $Q_{ij}(s)$ would represent the weight of a directed edge from node j to node i; the fact $Q_{ij}(s)$ is proper preserves the meaning of the directed edge as a *causal* dependency of y_i on y_j.
- Similarly, the entries of the matrix $[P(s)+(I-Q(s))D_1]$ carry the interpretation of causal weights characterizing the dependency of entires of y_1 on the m inputs, u. Note that when $D_1 = 0$, this matrix reduces to $P(s)$, which has *strictly* proper entries.

This leads naturally to the definition of signal structure.

Definition 5: The *signal structure* of a system G, with realization (1) and with dynamical structure function $(Q(s),P(s))$ characterized by (3), is a graph $\mathscr{W}$, with a vertex set $V(\mathscr{W})$ and edge set $E(\mathscr{W})$ given by:

- $V(\mathscr{W}) = \{u_1,...,u_m,y_{11},...,y_{1p_1},y_{21},...,y_{2p_2}\}$, each representing a manifest signal of the system, and
- $E(\mathscr{W})$ has an edge from u_i to y_{1j}, u_i to y_{2j}, y_{1i} to y_{1j} or y_{1i} to y_{2_j} if the associated entry in $[P(s)+(I-Q(s))D_1]$, D_2, $Q(s)$, or C_{21} (as given in Equation (3)) is nonzero, respectively.

We label the nodes of $V(\mathscr{W})$ with the name of the associated variable, and the edges of $E(\mathscr{W})$ with the associated transfer function entry from Equation (3).

Signal structure is fundamentally a different *type* of graph than either the computational or subsystem structure of a system because, unlike these other graphs, vertices of a system's signal structure represent *signals* rather than systems. Likewise, the edges of $\mathscr{W}$ represent *systems* instead of signals, as opposed to $\mathscr{C}$ or $\mathscr{S}$. We highlight these differences by using circular nodes in $\mathscr{W}$, in contrast to using square nodes for the vertices in $\mathscr{C}$ or $\mathscr{S}$.

D. Input Output Sparsity Structure

Another notion of structure exhibited by a system is the pattern of zeros portrayed in its transfer function matrix, where "zero" refers to the value of the particular transfer function element, not a transmission zero of the system. Like signal structure, this type of structure is particularly meaningful for multiple-input multiple-output systems, and, like signal structure, the corresponding graphical representation reflects the dependance of system output variables on system input variables. Thus, vertices of the graph will be signals, represented by circular nodes, and the edges of the graph will represent systems, labeled with the corresponding transfer function element; a zero element thus corresponds to the absence of an edge between the associated system input and output. Formally, we have the following definition

Definition 6: The *input output sparsity structure of the transfer function* of a system G is a graph $\mathscr{Z}$, with a vertex set $V(\mathscr{Z})$ and edge set $E(\mathscr{Z})$ given by:

- $V(\mathscr{Z}) = \{u_1,...,u_m,y_1,...,y_p\}$, each representing a manifest signal of the system, and
- $E(\mathscr{Z})$ has an edge from u_i to y_j if G_{ji} is nonzero.

We label the nodes of $V(\mathscr{Z})$ with the name of the associated variable, and the edges of $E(\mathscr{Z})$ with the associated element from the transfer function $G(s)$.

Unlike signal structure, note that the sparsity structure of the transfer function matrix describes the closed-loop dependency of an output variable on a particular input variable, not its *direct* dependence. As a result, the graph is necessarily bipartite, and all edges will begin at an input node and terminate at an output node; no edges will illustrate dependencies between output variables.

III. Relationships between Representations of Structure

In this section, we explore the relationships between the four representations of structure defined above. What we find is that some representations of structure are more informative than others. We also explore how signal and subsystem structure encode fundamentally different types of structural information. We illustrate these differences with an examples.

Different system representations portray different aspects of system structure. For example, the complete computational structure details the structural dependencies among fundamental units of computation (components of the manifest variables u,y or the LTI vector functions f,g,h in (1)). Using complete computational structure to model system structure requires knowledge of the parameters associated with each fundamental unit of computation. Signal and subsystem structure do not require knowledge of such details in their description. As a condensation graph, subsystem structure essentially condenses fundamental units of computation to form subsystems and models the closed-loop transfer function of each subsystem. Signal structure models the SISO transfer functions describing direct causal dependencies between some of the outputs and inputs of the fundamental units of computation, namely those that are manifest variables. Sparsity structure models the closed-loop dependencies of system outputs on inputs. Thus, complete computational structure appears to be the most demanding or information-rich description of system structure. This intuition is made precise with the following result (see [13] for proof):

Theorem 1: Suppose a complete computational structure has minimal intricacy realization (A_o, B_o, C_o, D_o) with

$$C_0 = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

and C_{11} invertible. Then the complete computational structure specifies a unique subsystem, signal, and sparsity structure.

It is well known that a transfer function $G(s)$ can be realized using an infinite number of state-space realizations. Without additional assumptions, e.g. full state feedback, it is impossible to uniquely associate a single state-space realization with a given transfer function. On the other hand, a state space realization specifies a unique transfer function. In this sense, a transfer function contains less information than the state space realization.

Similarly, subsystem, signal, and sparsity structure can be realized using multiple complete computational structures. Without additional assumptions, it is impossible to associate a unique complete computational structure with a given subsystem, signal, or sparsity structure. Theorem 1 shows that a complete computational structure specifies a unique subsystem, signal, and sparsity structure. In this sense, a complete computational structure is a more informative description of system structure than subsystem, signal and sparsity structure. The next result is similar and follows directly from the one-to-one correspondence of a system's transfer function with its sparsity structure.

Theorem 2: Every subsystem structure or signal structure specifies a unique sparsity structure.

Subsystem structure and signal structure are fundamentally different descriptions of system structure. In general, subsystem structure does not encapsulate the information contained in signal structure. Signal structure describes direct causal dependencies between manifest variables of the system. Subsystem structure describes closed loop dependencies between manifest variables involved in the interconnection of subsystems. Both representations reveal different perspectives of a system's structure. The next result makes this relationship between subsystem and signal structure precise.

Theorem 3: Given a system G, let $\mathscr{F}(N,S)$ be the LFT representation of a subsystem structure $\mathscr{S}$. In addition, let the signal structure of the system G be denoted as in equation (3). Let $Y(S_i)$ denote the outputs associated with subsystem S_i. Define $[Q_{int}(s)]_{ij} \equiv \begin{cases} \bar{Q}_{ij}(s) & y_i, y_j \in Y(S_k),\ S_k \in V(\mathscr{S}) \\ 0 & \text{otherwise,} \end{cases}$

and $Q_{ext} \equiv \bar{Q}(s) - Q_{int}(s)$. Then the signal structure and subsystem structure are related in the following way:

$$S \begin{bmatrix} L & K \end{bmatrix} = (I - Q_{int})^{-1} \begin{bmatrix} \bar{P} & Q_{ext} \end{bmatrix} \tag{4}$$

Proof: Examining relation (4), observe that the ij^{th} entry of the left hand side describes the closed loop causal dependency from the j^{th} entry of $\begin{bmatrix} U^T & Y^T \end{bmatrix}^T$ to Y_i. By closed loop, we mean that they do not describe the internal dynamics of each subsystem, e.g. the direct causal dependencies among outputs of a single subsystem. Thus, these closed loop causal dependencies are obtained by solving out the intermediate direct causal relationships, i.e. the entries in Q_{int}. Notice that the right hand side of (4) also describes the closed loop map from $\begin{bmatrix} U^T & Y^T \end{bmatrix}^T$ to Y, and in particular the ij^{th} entry of $(I - Q_{int})^{-1} \begin{bmatrix} \bar{P} & Q_{ext} \end{bmatrix}$ describes the closed loop causal dependency from the jth entry of $\begin{bmatrix} U & Y \end{bmatrix}^T$ to Y_i. ■

As a special case, notice that for single output subsystem structures, Q_{int} becomes the zero matrix and that for subsystem structures with a single subsystem, S becomes the system transfer function, L becomes the identity matrix, $Q_{int} = \bar{Q}$, and Q_{ext} and K are both zero matrices, thus specializing to Lemma 1 in [5] . The primary import of this result is that a single subsystem structure can be consistent with two or more signal structures and that a single signal structure can be consistent with two or more subsystem structures. We illustrate the latter scenario here with a simple example, a more intricate example of the former scenario is found in [13].

Example 1: A Signal Structure consistent with two Subsystem Structures

In this example, we will show how a signal structure can be consistent with two subsystem structures. To do this we construct two different generalized state-space realizations that yield the same minimal intricacy realization but different admissible partitions. The result is two different subsystem structures that are consistent with the same signal structure. First, we consider the complete computational structure $\mathscr{C}_1$ with generalized state-space realization

$$\left(\begin{bmatrix} A_1 & \hat{A}_1 \\ \bar{A}_1 & \tilde{A}_1 \end{bmatrix}, \begin{bmatrix} B_1 \\ \bar{B}_1 \end{bmatrix}, \begin{bmatrix} C_1 & \bar{C}_1 \end{bmatrix}, D_1 \right) \tag{5}$$

where

$$A_1 = \begin{bmatrix} -4 & 1 & 0 & 0 & 1 \\ 1 & -7 & 0 & 0 & 3 \\ 0 & 0 & -6 & 0 & 0 \\ 0 & 0 & 0 & -6 & 0 \\ 1 & 2 & 0 & 0 & -10 \end{bmatrix}, \quad \hat{A}_1 = \begin{bmatrix} 0 & 0 & 2 & 1 \\ 0 & 0 & 2 & 1 \\ 2 & 1 & 0 & 1 \\ 1 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\bar{A}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \qquad \tilde{A}_1 = \mathbf{0_{4x4}},$$

$$B_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T,\ \bar{B}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T,$$
$$C_1 = \mathbf{0_{4x5}}, \qquad \bar{C}_1 = \mathbf{I}_4,$$

and $D_1 = \mathbf{0_{4\times1}}$. Next consider the complete computational structure $\mathscr{C}_2$ with generalized state-space realization

$$\left(\begin{bmatrix} A_2 & \hat{A}_2 \\ \bar{A}_2 & \tilde{A}_2 \end{bmatrix}, \begin{bmatrix} B_2 \\ \bar{B}_2 \end{bmatrix}, \begin{bmatrix} C_2 & \bar{C}_2 \end{bmatrix}, D_2 \right) \tag{6}$$

where

$$A_2 = \begin{bmatrix} -4 & 1 & 0 & 0 & 1 \\ 1 & -7 & 0 & 0 & 3 \\ 0 & 0 & -6 & 1 & 0 \\ 0 & 0 & 2 & -6 & 0 \\ 1 & 2 & 0 & 0 & -10 \end{bmatrix} \quad \hat{A}_2 = \begin{bmatrix} 0 & 0 & 2 & 1 \\ 0 & 0 & 2 & 1 \\ 2 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$
$$\bar{A}_2 = \bar{A}_1 \qquad \tilde{A}_2 = [0]_4,$$
$$B_2 = B_1 = \mathbf{1_{5x1}}, \qquad \bar{B}_2 = \bar{B}_1 = \mathbf{0_{4x1}},$$
$$C_2 = C_1, \qquad \bar{C}_2 = \bar{C}_1,$$

and $D_2 = D_1$. The difference between these two computational structures is evident more in the subsystem structure representation of the system - note how replacing A_1 with A_2, essentially internalizes manifest dynamics. The result is that $\mathscr{C}_2$ admits a subsystem structure $\mathscr{S}_2$ which condenses two of the subsystems of $\mathscr{S}_1$ into a single subsystem.

We draw $\mathscr{S}_1$ and $\mathscr{S}_2$ in Figures 1(a) and 1(b) respectively. The LFT representation of $\mathscr{S}_1$ is given by $\mathscr{F}(N_1, S_1)$ with

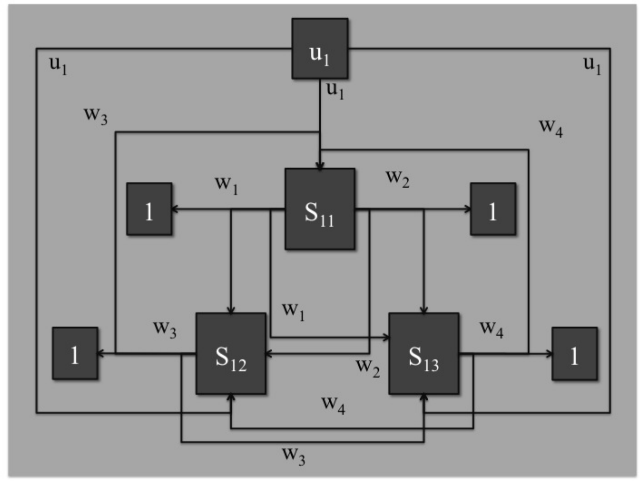
$$N_1 = \left[\begin{array}{c|c} \mathbf{0_{4x1}} & \mathbf{I_4} \\ \hline 0 & 0\ 0\ 1\ 0 \\ 0 & 0\ 0\ 0\ 1 \\ 1 & 0\ 0\ 0\ 0 \\ 0 & 1\ 0\ 0\ 0 \\ 0 & 0\ 1\ 0\ 0 \\ 0 & 0\ 0\ 0\ 1 \\ 1 & 0\ 0\ 0\ 0 \\ 0 & 1\ 0\ 0\ 0 \\ 0 & 0\ 1\ 0\ 0 \\ 0 & 0\ 0\ 1\ 0 \\ 1 & 0\ 0\ 0\ 0 \end{array}\right] \quad S_1 = \begin{bmatrix} S_{11} & 0 & 0 \\ 0 & S_{12} & 0 \\ 0 & 0 & S_{13} \end{bmatrix},$$

with S_{11}, S_{12}, S_{13} given by

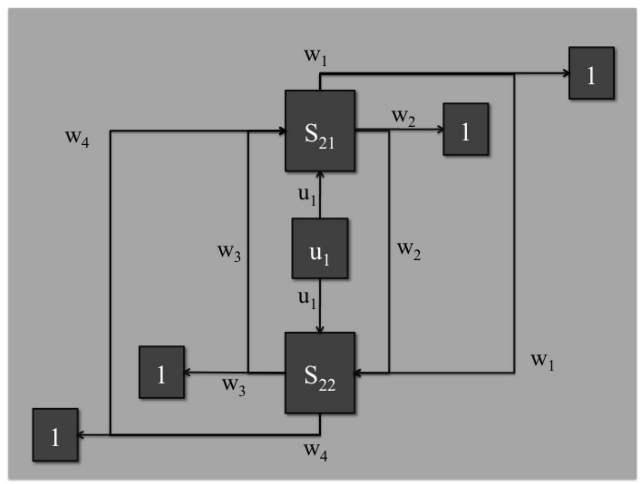
$$\begin{bmatrix} \frac{2(s^2+18s+76)}{s^3+21s+130s+234} & \frac{s^2+18s+76}{s^3+21s+130s+234} & \frac{s^2+19s+86}{s^3+21s^2+130s+234} \\ \frac{2(s^2+15s+52)}{s^3+21s^2+130s+234} & \frac{s^2+15s+52}{s^3+21s^2+130s+234} & \frac{(13+s)(s+5)}{s^3+21s^2+130s+234} \end{bmatrix},$$
$$\begin{bmatrix} \frac{2}{s+6} & \frac{1}{s+6} & \frac{1}{s+6} & \frac{1}{s+6} \end{bmatrix},$$
$$\begin{bmatrix} \frac{1}{s+6} & \frac{2}{s+6} & \frac{2}{s+6} & \frac{1}{s+6} \end{bmatrix}$$

respectively. The LFT representation of $\mathscr{S}_2$ is represented as the LFT $\mathscr{F}(N_2, S_2)$ where

$$N_2 = \left[\begin{array}{c|c} \mathbf{0_{4\times1}} & \mathbf{I_4} \\ \hline 0 & 0\ 0\ 1\ 0 \\ 0 & 0\ 0\ 0\ 1 \\ 1 & 0\ 0\ 0\ 0 \\ 0 & 1\ 0\ 0\ 0 \\ 0 & 0\ 1\ 0\ 0 \\ 1 & 0\ 0\ 0\ 0 \end{array}\right], \quad S_2 = \begin{bmatrix} S_{21} & 0 \\ 0 & S_{22} \end{bmatrix}, \text{ and}$$



(a) The subsystem structure $\mathscr{S}_1$ of system (5) with complete computational structure $\mathscr{C}_1$. The vertices of subsystem and complete computational structure always represent systems while edges represent system variables.



(b) The subsystem structure $\mathscr{S}_2$ of system (6) with complete computational structure $\mathscr{C}_2$. Notice the feedback loop between subsystem $S_{22}(s)$ and $S_{21}(s)$.

Fig. 1. The subsystem structures $\mathscr{S}_1$ and $\mathscr{S}_2$ are generated from almost identical systems (5) and (6). Specifically, the feedback between S_{12} and S_{13} is a manifest interconnection involving the manifest variables w_3 and w_4; thus it is included in $\mathscr{S}_1$ (Figure 1(a)). The same feedback dynamics are internalized in (6) so that the corresponding subsystem structure $\mathscr{S}_2$ (Figure 1(b)) only models the interconnection between two subsystems.

$$S_{21} = \begin{bmatrix} \frac{2(s^2+18s+76)}{s^3+21s^2+130s+234} & \frac{s^2+18s+76}{s^3+21s^2+130s+234} & \frac{s^2+19s+86}{s^3+21s^2+130s+234} \\ \frac{2(s^2+15s+52)}{s^3+21s^2+130s+234} & \frac{s^2+15s+52}{s^3+21s^2+130s+234} & \frac{(13+s)(s+5)}{s^3+21s^2+130s+234} \end{bmatrix},$$
$$S_{22} = \begin{bmatrix} \frac{2s+13}{s^2+12s+34} & \frac{s+8}{s^2+12s+34} & \frac{7+s}{s^2+12s+34} \\ \frac{s+10}{s^2+12s+34} & \frac{2(7+s)}{s^2+12s+34} & \frac{s+8}{s^2+12s+34} \end{bmatrix}.$$

However, if we consider the minimal intricacy realizations of $\mathscr{C}_1, \mathscr{C}_2$ we get the same state-space realization (A_o, B_o, C_o, D_o) with

$$A_o = \begin{bmatrix} -4 & 1 & 2 & 1 & 1 \\ 1 & -7 & 2 & 1 & 3 \\ 2 & 1 & -6 & 1 & 0 \\ 1 & 2 & 2 & -6 & 0 \\ 1 & 2 & 0 & 0 & -10 \end{bmatrix}, \quad B_o = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

and $C_o = \begin{bmatrix} I_4 & \mathbf{0_{4\times 1}} \end{bmatrix}$. The signal structure (see Figure III) of the system is thus specified by the dynamical structure function $(Q,P)(s)$, with

$$Q(s) = \begin{bmatrix} 0 & \frac{12+s}{s^2+14s+39} & \frac{2(s+10)}{s^2+14s+39} & \frac{s+10}{s^2+14s+39} \\ \frac{13+s}{s^2+17s+64} & 0 & \frac{2(s+10)}{s^2+17s+64} & \frac{s+10}{s^2+17s+64} \\ \frac{2}{s+6} & \frac{1}{s+6} & 0 & \frac{1}{s+6} \\ \frac{1}{s+6} & \frac{2}{s+6} & \frac{2}{s+6} & 0 \end{bmatrix}$$

$$P(s) = \begin{bmatrix} \frac{11+s}{s^2+14s+39} \\ \frac{13+s}{s^2+17s+64} \\ \frac{1}{s+6} \\ \frac{1}{s+6} \end{bmatrix}$$

■

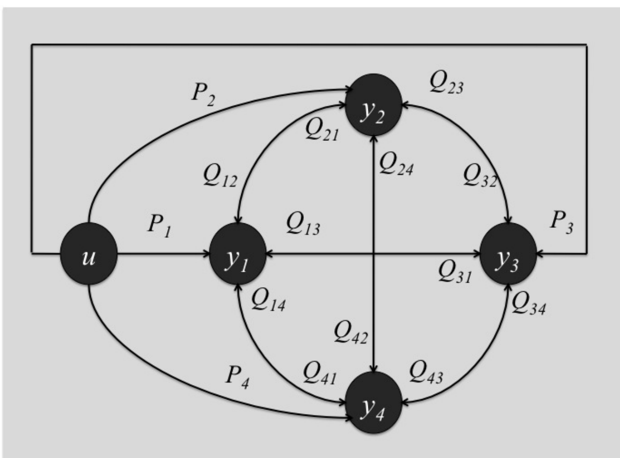


Fig. 2. The signal structure of both system (5) and (6). $\mathscr{S}_1$ and $\mathscr{S}_2$ are distinct subsystem structures generated from these two distinct generalized state-space realizations but they are consistent with same signal structure.

IV. CONCLUSION AND FUTURE WORK

We have briefly defined four definitions of structure: complete computational, subsystem, signal, and sparsity structure. Each definition is a graphical representation of an LTI input-output system's structure. We also reviewed some ways of mathematically describing these representations of structure. Using these mathematical representations, we derived some of the relationships between these different notions of structure. Our results elucidate the type of information present in each representation of structure and in the case of systems with single output structure (see [13] for discussion), specify a ranking of information content from most informative to least informative. For example, we found that complete computational structure, subsystem structure, and signal structure specify a unique sparsity structure.

These kinds of results opens the door for new research problems. For example, we pose the question of realizing a subsystem structure from a sparsity structure, or a complete computational structure from a given sparsity structure. From this point, we can proceed to consider the question of minimal realization which requires an appropriate definition of structural complexity. Exploring how to appropriately define the complexity of a subsystem or signal structure is in itself an open research problem. Thus, future work [13] will explore the resulting research problems that arise from the results in this paper, [7], [6] and the framework provided in [9]. At a more general level, future research will also investigate different representations of structure and their relationships within the behavioral framework provided in [4].

V. ACKNOWLEDGMENTS

The authors gratefully acknowledge the generous support from the Air Force Research Laboratory Grant FA 8750-09-2-0219 and by the Engineering and Physical Sciences Research Council Grant EP/G066477/1.

REFERENCES

[1] R. Olfait-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays", *IEEE Transactions on Automatic Control*, vol. 82, no. 7, pp. 1520–1533, September 2004.

[2] S. Roy, Y. Wan, and A. Saberi, "On time scale design for networks", *International Journal of Control*, vol. 49, no. 9, pp. 1313–1325, July 2009.

[3] Henrik Sandberg and Richard M. Murray, "Model reduction of interconnected linear systems", *Optimal Control; Application and Methods, Special Issue on Directions, Applications, and Methods in Robust Control*, 2009, Preprint at `http://www.ee.kth.se/~hsan/intermodred.pdf`.

[4] J. C. Willems, "The behavioral approach to open and interconnected systems", *Control Systems Magazine*, vol. 27, no. 6, pp. 46–99, 2007.

[5] J. Gonçalves and S. Warnick, "Dynamical structure functions for the reverse engineering of LTI networks", *IEEE Transactions of Automatic Control, 2007*, August 2007.

[6] Yuan Y., Stan G.B., Warnick S., and Gonalves J., "Minimal dynamical structure realisations with application to network reconstruction from data", in *Proceedings of the 48th IEEE Conference on Decision and Control (IEEE-CDC 2009)*, December 2009.

[7] E. Yeung, J. Gonçalves, H. Sandberg, and S. Warnick, "Network structure preserving model reduction with weak a priori structural information", *Proceedings of 2009 Conference on Decision and Control*, December 2009.

[8] R. Howes, S. Warnick, and J. Gonçalves, "Dynamical structure analysis of sparsity and minimality heuristics for reconstruction of biochemical networks", *Proceedings of the Conference on Decision and Control*, December 2008.

[9] E. Yeung, J. Goncalves, H. Sandberg, and S. Warnick, "Representing structure in linear interconnected dynamical systems", *The Proceedings of the 49th IEEE Conference on Decision and Control to appear*, December 2010, Preprint at `http://idealabs.byu.edu/publications/CDC10.pdf`.

[10] K. Schmidt, *Dynamical Systems of Algebraic Origin*, Basel, Boston, 1943.

[11] K. Zhou J. Doyle, K. Glover, *Robust and Optimal Control*, Prentice Hall, Englewood Cliffs, N.J., 1996.

[12] C. Ward, E. Yeung, T. Brown, B. Durtschi, S. Weyerman, R. Howes, J. Goncalves, H. Sandberg, and S. Warnick., "A comparison of network reconstruction methods for chemical reaction networks", *Proceedings of the Foundations for Systems Biology and Engineering Conference*, August 2009.

[13] E. Yeung, H. Sandberg, J. Gonçalves, and S. Warnick, "The meaning of structure in dynamic systems", *Special Issue of the Control Systems Magazine, to appear*, July 2010.