49th IEEE Conference on Decision and Control
December 15-17, 2010
Hilton Atlanta Hotel, Atlanta, GA, USA

Representing Structure in Linear Interconnected Dynamical Systems

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

Abstract— **Interconnected dynamical systems are a pervasive component in our modern world's infrastructure. One of the fundamental steps to understanding the complex behavior and dynamics of these systems is determining how to appropriately represent their structure. In this work, we discuss different ways of representing a system's structure. We define and present, in particular, four representations of system structure–complete computational, subsystem, signal, and zero pattern structure–and discuss some of their fundamental properties. We illustrate their application with a numerical example and show how radically different representations of structure can be consistent with a single LTI input-output system.**

I. Introduction

Dynamical systems can be useful for modeling complex phenomena with underlying network structure. Examples of such phenomena include the power grid, formations of vehicles, distributed systems, social networks, chemical reaction networks, and cooperative multi-agent systems. Depending on the context, the notion of a system's structure can lead to a variety of interpretations. Even within a single application, a dynamic system's structure can be discussed in multiple ways.

For example, in chemical reaction networks, system structure can refer to the direct physical interactions between chemical species of the system, e.g. binding events, allosteric regulation. On the other hand, system structure in chemical reaction networks can also refer to nonphysical correlations or dependencies among signals (manifest variables). This notion of structure may be relevent when 1) only a subset of the chemical species can be directly measured, 2) the number of different chemical species in the system is so large that only a subset of chemical species can be modeled to ensure computational tractability. In this context, recent research [1], [2] has represented system structure as the causal dependencies between manifest variables.

Zooming out to the microscopic level, system structure can refer to the interaction between multiple chemical reaction networks physically separated by selectively permeable membranes, e.g. a cluster of cells interacting via intercellular cross-talk. In such systems direct physical interaction may not be an appropriate basis for formulating system structure; the more appropriate representation of system structure describes interaction between subsystems.

One of the most important issues, in characterizing a system's structure, is understanding the available representations for studying system structure and the relationships between these different representations. Our goal is to build a framework for understanding the different ways to represent or describe system structure. We will define four definitions of system structure which reflect different aspects of a system's structure. In Section II we consider the most complete representation of system structure: complete computational structure. In Section III we introduce the concept of a partial representation of structure– these representations will highlight certain aspects of system structure while obscuring unnecessary or unwanted complexity. Throughout these two sections, we will show how each of these four definitions defines a dynamical graph with an associated set of dynamics. Finally, in Section IV, we conclude with a numerical example to show how each of these representations highlight different aspects of a dynamical system's structure.

II. Complete Computational Structure

The complete computational structure of a system characterizes the actual processes it uses to sense properties of its environment, represent and store variables internally, and affect change externally. At the core of these processes are information retrieval issues such as the encoding, storage, and decoding of quantities that drive the system's dynamics.

Mathematically, state equations are typically used to describe these mechanisms. Although there may be many realizations that describe the same input-output properties of a particular system, its complete computational structure is the architecture of the *particular realization* fundamentally used to store state variables in memory and transform system inputs to the corresponding outputs.

To make this concept precise, we begin by considering a linear system G with state space realization:

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

where $A\in\mathbb{R}^{n\times n}$, $\hat{A}\in\mathbb{R}^{n\times l}$, $\bar{A}\in\mathbb{R}^{l\times n}$, $\tilde{A}\in\mathbb{R}^{l\times l}$, $B\in\mathbb{R}^{n\times m}$, $\bar{B}\in\mathbb{R}^{l\times m}$, $C\in\mathbb{R}^{p\times n}$, $\bar{C}\in\mathbb{R}^{p\times l}$, and $D\in\mathbb{R}^{p\times m}$ and $u\in\mathscr{R}^m$, $x\in\mathbb{R}^n$, $w\in\mathbb{R}^l$, and $y\in\mathbb{R}^p$ so that m, n, l and p characterize the dimensions of the input, state, auxiliary and output vectors, respectively.

The auxiliary variables, w, are used to characterize intermediate computation in the composition of functions. Thus, for example, we distinguish between $\dot{x}_1 = f_1(x) = -x_1$ and $\dot{x}_1 = f_1(x) = -2(.5x_1)$ by computing the latter as $\dot{x}_1 = -2w_1$ and $w_1 = g_1(x) = .5x_1$. In this way, the auxiliary variables serve to identify stages in the computation of the state space realization (1). Frequently we may not require any auxiliary variables in our description of the system; indeed it is the standard practice to eliminate auxiliary variables to simplify the state descriptions of systems, see [3], [4] for an extensive discussion. Nevertheless, as we discuss structure, it will be critical to use auxiliary variables to distinguish between systems with dynamically equivalent, yet structurally distinct architectures, leading to the following definition.

Definition 1: Given a system (1), we call the number of auxiliary variables, l, the *intricacy* of the realization.

Notice that by eliminating these auxiliary variables, we obtain the standard linear time invariant state space realization. We note with the following lemma that if $I-\tilde{A}$ is invertible, then elimination is always possible.

Lemma 1: For any system (1) with intricacy $l > 0$ and $(I-\tilde{A})$ invertible, there exists a unique *minimal intricacy* realization (A_o, B_o, C_o, D_o) with $l = 0$ such that for every solution $(u(t), x(t), w(t), y(t))$ of (1), $(u(t), x(t), y(t))$ is a solution of (A_o, B_o, C_o, D_o).

To understand the structure of (1), we need a notion of dependence of a function on its arguments. For example, the function $f(x,y,z) = xy - x + z$ clearly depends on z, but it only depends on x when $y \neq 1$ (or on y when $x \neq 0$). Since "structure" refers at some level to the dependence of the system variables on each other, it is important that our notion of dependence be made clear.

Definition 2: A function $f(w)$, from l-dimensional domain $\mathscr{W}$ to s-dimensional co-domain $\mathscr{Z}$, is said to *depend* on the i^{th} variable, w_i, if there exist values of the other $l-1$ variables w_j, $j \neq i$, such that $f(w)$ is not constant over all values of w_i while holding the others variables fixed. If $l = 1$, then $f(w)$ *depends* on w if it is not constant over all values of w.

Definition 3: Given a system G with realization (1), its *complete 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, each element associated with one mechanism that produces either an input, state, auxiliary variable, or output 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.

The complete computational structure of a system is a graphical representation of the dependency among input, state, auxiliary, and output variables that is in direct, one-to-one correspondence with the system's state realization, generalized to explicitly account for composition intricacy. All structural and behavioral information is fully represented by this description of a system. Nevertheless, this representation of the system can also be unwieldy for large systems with intricate structure.

III. Partial Structure Representations

Complex systems are often characterized by intricate computational structure and complicated dynamic behavior. State descriptions and their corresponding complete computational structures accurately capture both the system's structural and dynamic complexity, nevertheless these descriptions themselves can be too complicated to convey an efficient understanding of the nature of the system. Simplified representations are then desirable.

One way to simplify the representation of a system is to restrict the structural information of the representation while maintaining a complete description of the system's dynamics. An extreme example of this type of simplified representation is the transfer function of a linear time invariant (LTI) system. A transfer function completely specifies the system's input-output dynamics without retaining any information about the computational structure of the system.

We use this power of a transfer function to obfuscate structural information to develop three distinct partial structure representations of an LTI system: subsystem structure, signal structure, and the zero pattern of a (multiple input, multiple output) system's transfer function matrix.

A. Subsystem Structure

One of the most natural ways to reduce the structural information in a system's representation is to partition the nodes of its computational structure into subsystems, then replace these subsystems with their associated transfer function. Each transfer function obfuscates the structure of its associated subsystem, and the remaining (partial) structural information in the system is the interconnection between transfer functions.

Subsystem structure refers to the interconnection structure of the subsystems of a given system. 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. Such abstractions have been used in various ways to simplify the structural descriptions of complex systems [5], [6], for example by "condensing" strongly connected components or other groups of vertices of a graph into single nodes, but here we define a *particular* condensation graph as the subsystem structure of the system. We begin by characterizing the partitions of $\mathscr{C}$ that identify subsystems.

Definition 4: 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 $V(\mathscr{C})$.

Definition 5: 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 highest 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 $E(\mathscr{C})$.

A system's subsystem structure always exists, although it may be trivial (a single internal block) for those systems that do not decompose naturally into an interconnection of subsystems. Note that $\mathscr{S}$ always identifies the most refined subsystem structure possible; it is by defining $\mathscr{S}$ as the most refined subsystem structure, i.e. requiring $V(\mathscr{S})$ to have maximal cardinality, that guarantees the uniqueness and well-definedness of a system's subsystem structure. We state this as a lemma and refer the reader to [4] for a proof.

Lemma 2: Given a system G with realization (1) and associated computational structure $\mathscr{C}$, the system's subsystem structure $\mathscr{S}$ exists and is unique.

The subsystem structure reveals the way natural subsystems are interconnected, and it can be represented in other ways besides (but equivalent to) identifying $\mathscr{S}$. For example, one common way to identify this kind of subsystem structure is to write the system as the linear fractional transformation (LFT) with a block diagonal "subsystem" component and a static "interconnection" component (see [7], [8], [9] on using the LFT to represent structure). In general, the LFT associated with $\mathscr{S}$ will have the form

$$N = \begin{bmatrix} 0 & I \\ L & K \end{bmatrix} \quad S = \begin{bmatrix} S_1 & 0 & \dots \\ 0 & \ddots & \\ \vdots & 0 & S_q \end{bmatrix} \tag{2}$$

where q is the number of distinct subsystems, and L and K are each binary matrices of the appropriate dimension. In this work we will restrict our attention to where $\begin{bmatrix} L & K \end{bmatrix}$ is a permutation matrix. 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, however, N is static and L and K are binary matrices. Notice that there is no loss of generality in this restriction on N, since every LFT with a dynamic $N(s)$ can be rewritten so that all dynamics are expressed in the S matrix. This convention simply allows for a unique LFT (up to a reordering of the manifest variables) to be associated with each subsystem structure $\mathscr{S}$.

B. Signal Structure

Another very natural way to partially describe the structure of a system is to characterize the causal dependence among each of its manifest variables. Although subsystem structure also considers causal dependencies between manifest variables, signal structure will not demand that the nodes of $\mathscr{C}$ be *partitioned*, and thus it offers a perspective on the dependency between manifest signals without characterizing the internal interconnection of subsystems.

We begin by considering the system (A,B,C,D) given by

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \\ \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} &= \begin{bmatrix} I & 0 \\ C_{21} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} D_1 \\ D_2 \end{bmatrix} u \end{aligned} \tag{3}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $y_1 \in \mathbb{R}^{p_1}$ with $p_1 \le n$, and $y_2 \in \mathbb{R}^{p_2}$ with $p_1 + p_2 = p$. The next lemma ensures that restricting our attention to systems of this form does not result in a loss of generality; note that (A,B,C,D) in (3) are not the same (A,B,C,D) from (1), although no confusion (and significantly simpler notation) should arise as the context makes each reference clear.

Lemma 3: Every system (1), with minimally intricate realization (A_o,B_o,C_o,D_o), has a realization of the form (3), where $p_1 \le n$ is the rank of C_o, and the order of the outputs, y, may be renumbered.

Proof: Consider the zero-intricacy realization, given by (A_o,B_o,C_o,D_o) and which Lemma 1 ensures is well defined, of any system characterized by (1). Let $p_1 \le n$ be the rank of C_o, and let P_L be a permutation such that the first p_1 rows of $P_L C_o$ are linearly independent. Now, consider a permutation P_R, so that the first p_1 columns of $C_o P_R$ are linearly independent, resulting in the corresponding partition

$$P_L C_o P_R = \begin{bmatrix} C_1 & C_2 \\ C_3 & C_4 \end{bmatrix}. \tag{4}$$

Note that $C_1 \in \mathbb{R}^{p_1 \times p_1}$ is invertible, leading to the transformation

$$T = \begin{bmatrix} C_1^{-1} & -C_1^{-1} C_2 \\ 0 & I \end{bmatrix}. \tag{5}$$

We then see that

$$P_L C_o P_R T = \begin{bmatrix} I & 0 \\ C_{21} & 0 \end{bmatrix}, \tag{6}$$

with $C_{21} = C_3 C_1^{-1}$. Thus a state transformation $\bar{x} = (P_R T)^{-1} x$, along with a renumbering of the system outputs consistent with P_L, will yield a realization of the form (3). ∎

We thus characterize the signal structure of any system (1) in terms of the dynamical structure function of the canonical realization (3). Dynamical structure functions were defined in [1], [10], but we derive them here again to apply them to general linear systems. Taking Laplace transforms of (3), and assuming initial conditions are zero, yields

$$\begin{bmatrix} sX_1 \\ sX_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} U$$

where $X(s)$ denotes the Laplace transform of $x(t)$, etc. Solving for X_2 in the second equation and substituting into the first then yields

$$sX_1 = W(s)X_1 + V(s)U \tag{7}$$

where $W(s) = \left[A_{11} + A_{12}(sI - A_{22})^{-1} A_{21}\right]$ and $V(s) = \left[B_1 + A_{12}(sI - A_{22})^{-1} B_2\right]$. Let $D(s)$ be the matrix of the diagonal entries of $W(s)$, yielding

$$X_1 = Q(s)X_1 + P(s)U \tag{8}$$

where $Q(s) = (sI - D(s))^{-1}(W(s) - D(s))$ and $P(s) = (sI - D(s))^{-1}V(s)$. From (3) we note that $X_1 = Y_1 - D_1 U$, which, substituting into (8), then yields:

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} Q(s) \\ C_{21} \end{bmatrix} Y_1 + \begin{bmatrix} P(s) + (I - Q(s))D_1 \\ D_2 - C_{21}D_1 \end{bmatrix} U \tag{9}$$

The matrices $(Q(s), P(s))$ are called the dynamical structure function of the system (3), and they characterize the dependency graph among manifest variables as indicated in Equation (9). 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 entries 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.

Definition 6: The *signal structure* of a system G, with realization (1) and equivalent realization (3) with dynamical structure function $(Q(s), P(s))$ characterized by (8), 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 variable 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 $[P(s) + (I - Q(s))D_1]$, $D_2 - C_{21}D_1$, $Q(s)$, or C_{21} (as given in Equations (8) and (9)) 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 (9).

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 in $\mathscr{C}$ or $\mathscr{S}$. We highlight these differences by using circular nodes in $\mathscr{W}$, in contrast to the square nodes in $\mathscr{C}$ or $\mathscr{S}$.

C. Zero Pattern Structure

A fourth natural notion of system structure is the description of output dependencies on input variables. In linear systems, these dependencies are characterized completely by the transfer function of the system. In particular, the location of zero and nonzero entries, or the *boolean structure*, of the transfer function matrix determine the dependency of each output on the input variables of the system. However, their characterization of input-output relationships is the most basic description of system structure.

Definition 7: The zero pattern structure of a MIMO system G, with realization (1) and transfer function $G(s)$ is a graph $\mathscr{Z}$, with 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,
- $E(\mathscr{Z})$ has an edge from u_j to y_i, if the associated ij^{th} entry in $[G(s)]$ 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 transfer function entry from the transfer function.

The zero pattern structure is the weakest description of system structure. For example, certain systems will exhibit dynamics where states combine to cancel out the effects of one or more inputs on an output. In these scenarios, a zero in the transfer function matrix is indicative of exact cancellation, as opposed to decoupling in the system. However, the zero pattern structure of a transfer function does not distinguish between these two situations. These cancelling dynamics are internal details that are too refined for the zero pattern to capture. In this way, the zero pattern is a partial representation of structure.

IV. Computational Example

In this section, we will study a LTI input-output system using all four representations of structure. These representations will illustrate that a system can simultaneously possess structure in a variety of ways. Begin with the linear system:

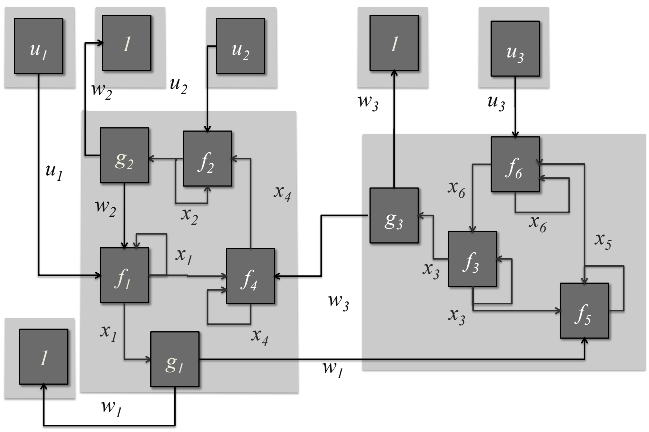
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} -3 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -5 & 0 & 0 & 5 \\ 1 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 9 & 0 & -12 & 0 \\ 0 & 0 & 0 & 0 & 7 & -7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} + \begin{bmatrix} 2 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

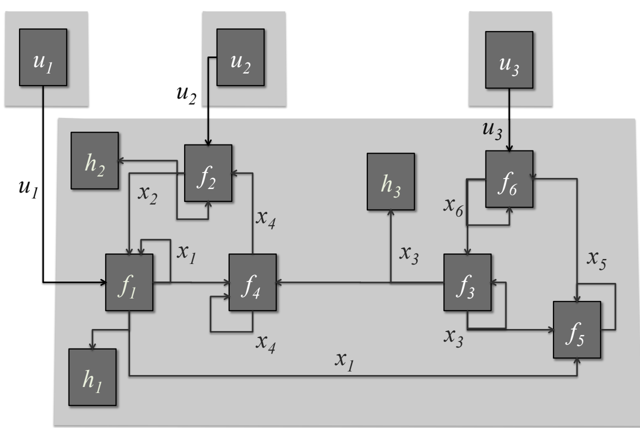
$$\begin{aligned} w &= \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix} x \\ y &= \begin{bmatrix} I_3 & 0_{3x6} \end{bmatrix} \begin{bmatrix} w \\ x \end{bmatrix} \end{aligned} \tag{10}$$

The computational structure of (10) is given in Figure 1(a). Notice that $\tilde{A}, \hat{B}, \bar{C}, D$ are zero matrices in this example. The auxiliary variables w_1, w_2, w_3 play a major role in characterizing the computational structure of (10). In particular, w_1 and w_3 act as linking variables between the vertex sets $\{f_1, f_2, f_4, g_2, g_1\}$ and $\{g_3, f_3, f_5, f_6\}$. Notice that w_1, w_2, w_3

are also manifest variables, as indicated by their connections to output vertices “1". The auxiliary variables keep track of the structural differences introduced by the composition of functions. This allows a degree of flexibility in how refined a view of the computational structure one considers “complete." If we eliminate these auxiliary variables, we get



(a) The computational structure $\mathscr{C}$ of a linear system with realization (10) and intricacy $l = 3$. Notice the system can be admissibly partitioned as two cyclic subsystems interconnected by manifest variables in a feedback loop.



(b) The complete computational structure $\mathscr{C}$ of realization (11) with minimal intricacy $l = 0$. Notice that in the absence of w_1, w_2, w_3, we lose the ability to admissibly partition $\mathscr{C}$ into two cyclic subsystems.

Fig. 1. The complete computational structure of realizations (10) and (11).

the minimal intricacy system:

$$\dot{x} = A_o x + B_o u \quad \text{and} \quad y = C_o x \tag{11}$$

where

$$A_o = \begin{bmatrix} -3 & 3 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -5 & 0 & 0 & 5 \\ 1 & 0 & 1 & -2 & 0 & 0 \\ 3 & 0 & 9 & 0 & -12 & 0 \\ 0 & 0 & 0 & 0 & 7 & -7 \end{bmatrix},$$

$$B_o = \begin{bmatrix} 2 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad C_o = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix}$$

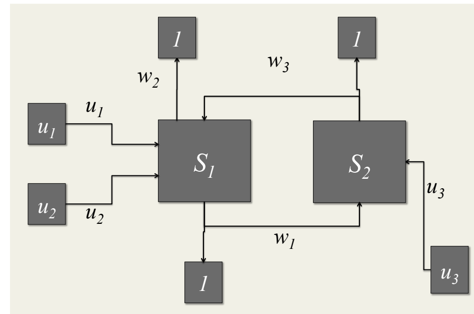


Fig. 2. The subsystem structure of $\mathscr{C}$ in Figure 1(a) with intricacy $l = 3$. Edges are labeled with manifest variables and vertices are labeled with transfer functions. Vertices carry the meaning of subsystems while edges carry the meaning of signals.

Notice that in the absence of auxiliary variables, $\mathscr{C}$ of the minimal intricacy realization has a different admissible partition than $\mathscr{C}$ of realization (10). This observation is important in determining the subsystem structure $\mathscr{S}$ of (10).

Recall that the subsystem structure is a condensation graph defined by the maximal admissible partition on $V(\mathscr{C})$. It is easy to see from the shaded regions in Figure 1(b) that if we omit all auxiliary variables (as in the minimal intricacy realization), the subsystem structure of (11) is comprised of a single subsystem with three inputs. If we retain the auxiliary variables, the subsystem structure $\mathscr{S}$ can be expressed as the LFT $\mathscr{F}(N,S)$, where N and the block diagonal entries S_1, S_2 of S are given by:

$$N = \left[\begin{array}{ccc|ccccc} e_1 & e_2 & e_3 & e_1 & e_5 & e_6 & e_3 & e_4 \end{array}\right]^T$$

$$S_1(s) = \begin{bmatrix} \frac{9}{2(s^3+6s+11s+3)} & \frac{-27(s+2)}{s^3+6s^2+11s+3} & \frac{6(s+1)(s+2)}{s^3+6s^2+11s+3} \\ \frac{3}{2(s+3)(s^3+6s+11s+3)} & \frac{-9(s^2+5s+6)}{s^3+6s^2+11s+3} & \frac{6}{s^3+6s^2+11s+3} \end{bmatrix}$$

$$S_2(s) = \begin{bmatrix} \frac{14}{5s^3+120s^2+895s+2037} & \frac{2(s+12)}{5s^3+120s^2+895s+2037} \end{bmatrix}$$

and satisfy the dynamics

$$\begin{bmatrix} Y \\ \pi(Y,U) \end{bmatrix} = N(s) \begin{bmatrix} U \\ Y \end{bmatrix}, \quad , \quad Y = S\pi \ ,$$

where $\pi = \begin{bmatrix} Y_3 & U_2 & U_1 & Y_1 & U_3 \end{bmatrix}^T$. Subsystem structure describes the interconnection between subsystems; it does not comment about the internal computational structure of individual subsystems. It is a graph connecting inputs, outputs, and subsystems.

To find the signal structure $\mathscr{W}$ of realization (10), we apply the state space transformation $z = Tx$ on the equivalent minimal intricacy realization, where T is the diagonal matrix $\text{diag}(3,2,10,1,1,1)$. The resulting realization is a simplified case of (3), with $D_1, D_2, C_{21} = 0$. Following the steps detailed in Subsection III-B and equations (7), (8), and (9), we derive the dynamical structure function (Q,P) where

$$Q(s) = \begin{bmatrix} 0 & \frac{s}{s+3} & 0 \\ \frac{1}{s^2+3s+2} & 0 & \frac{3}{2(s^2+3s+2)} \\ \frac{70s^2+1330s+5880}{s^5+43s^4+719s^3+5522s^2+17030s+8820} & 0 & 0 \end{bmatrix},$$

$$P(s) = \begin{bmatrix} \frac{6}{s+3} & 0 & 0 \\ 0 & \frac{-9}{s+1} & 0 \\ 0 & 0 & \frac{10s^2+190s+840}{s^4+31s^3+347s^2+1358s+735} \end{bmatrix}.$$

The entries of (Q,P) are transfer functions that characterize the causal dependencies between manifest variables. Figure 3 shows the signal structure of the system. Notice that in a signal structure graph, vertices represent signals/manifest variables and edges represent subsystems. The signal structure is essentially a single cycle graph with a feedback loop between Y_1 and Y_2, despite that it was derived from a system consisting of two cyclic subsystems. Nonetheless, these structures are related, as they both describe the system (10), but highlight different qualitative aspects of the system's structure.

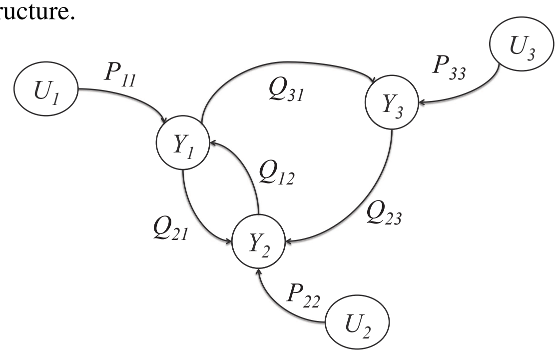


Fig. 3. The signal structure of system (10) is a single cyclic graph with one feedback loop between Y_1 and Y_2. Notice that this view of system structure obscures the subsystem structure of (10), namely the feedback interconnection of two internally cyclic subsystems.

Finally, the zero pattern structure of the system $\mathscr{Z}$ is illustrated in Figure 4. The associated transfer function of the system is a full matrix (this is the extreme case of a zero pattern structure with no zeros)

$$G = \begin{bmatrix} G_{11}(s) & G_{12}(s) & G_{13}(s) \\ G_{21}(s) & G_{22}(s) & G_{23}(s) \\ G_{31}(s) & G_{32}(s) & G_{33}(s) \end{bmatrix}.$$

Here, we have omitted the actual numerical entries since they are easily computed but cumbersome to write down.

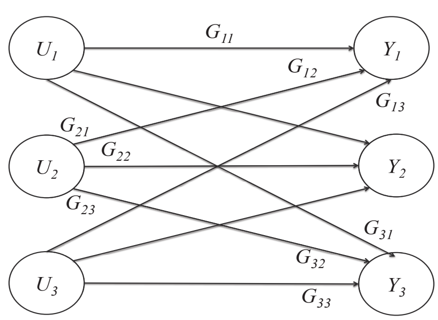


Fig. 4. The zero pattern structure $\mathscr{Z}$ of realization (10).

With the exception of edges linking outputs to outputs and edges linking inputs to inputs (which are never present in a zero pattern structure), every input has an outgoing edge to each output node. In this way, the zero pattern structure of realization (10) is fully connected and is the least structured or organized relative to its counterpart representations. It is possible to develop examples in which the computational, subsystem, and signal structures each in turn appear to be the least structured relative to the other three representations.

V. Conclusion

In conclusion, the complete computational, subsystem, signal, and zero pattern structure of a system each highlight different facets of system structure. A system can be highly organized with respect to its signal representation, but completely unstructured or typical with respect to its other three representations. We emphasize that our analysis has been concerned only with linear time invariant systems formulated within an input-output framework (though the notions of complete computational structure generalize to nonlinear models). This is because we have defined partial representations of structure using transfer functions, which are LTI input-output system constructs. Future research will investigate representing structure for more general systems, especially those described in the behavioral framework, see for example [3]. In addition, new problems in systems theory regarding subsystem, signal, and zero pattern sructure have arose such as structure realization, structure preserving model reduction, and approximate structure reconstruction. These research problems are described in [4] and will be the subject of future research as well.

VI. Acknowledgments

This work was supported in part by the Air Force Research Laboratory Grant FA 8750-09-2-0219 and by the Engineering and Physical Sciences Research Council grant number EP/G066477/1.

References

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

[2] Y. Y., S. G.B., W. S., and G. 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.

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

[4] E. Yeung, H. Sandberg, J. Gonçalves, and S. Warnick, "The meaning of structure in dynamic systems, *manuscript in preparation*," *Control Systems Magazine*, July 2010.

[5] D. Siljac, *Large Scale Dynamic Systems: Stability and Structure*. New York: North-Holland, 1978.

[6] F. Harary, *Graph Theory*. Reading, Massachusetts: Addison-Wesley Pub. Co., 1969.

[7] K. Z. J. Doyle, K. Glover, *Robust and Optimal Control*. Englewood Cliffs, N.J.: Prentice Hall, 1996.

[8] H. Sandberg and R. M. Murray, "Model reduction of interconnected linear systems," *Optimal Control Application and Methods*, 2009, preprint at `http://www.ee.kth.se/~hsan/intermodred.pdf`.

[9] A. Vandendorpe and P. Van Dooren, "On model reduction of interconnected systems," in *Proceedings International Symposium Math. Th. Netw. Syst.*, Belgium, 2004.

[10] 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.