2017-03-01

Theory and Applications of Network Structure of Complex Dynamical Systems

Vasu Nephi Chetty
Brigham Young University

Theory and Applications of Network Structure of Complex Dynamical Systems

Vasu N. Chetty

A dissertation submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Sean Warnick, Chair
Randal Beard
Daniel Zappala
Dennis Ng
Quinn Snell

Department of Computer Science

Brigham Young University

ABSTRACT

Theory and Applications of Network Structure
of Complex Dynamical Systems

Vasu N. Chetty
Department of Computer Science, BYU
Doctor of Philosophy

One of the most powerful properties of mathematical systems theory is the fact that interconnecting systems yields composites that are themselves systems. This property allows for the engineering of complex systems by aggregating simpler systems into intricate patterns. We call these interconnection patterns the "structure" of the system. Similarly, this property also enables the understanding of complex systems by decomposing them into simpler parts. We likewise call the relationship between these parts the "structure" of the system. At first glance, these may appear to represent identical views of structure of a system. However, further investigation invites the question: are these two notions of structure of a system the same?

This dissertation answers this question by developing a theory of dynamical structure. The work begins be distinguishing notions of structure from their associated mathematical representations, or models, of a system. Focusing on linear time invariant (LTI) systems, the key technical contributions begin by extending the definition of the dynamical structure function to all LTI systems and proving essential invariance properties as well as extending necessary and sufficient conditions for the reconstruction of the dynamical structure function from data. Given these extensions, we then develop a framework for analyzing the structures associated with different representations of the same system and use this framework to show that interconnection (or subsystem) structures are not necessarily the same as decomposition (or signal) structures. We also show necessary and sufficient conditions for the reconstruction of the interconnection (or subsystem) structure for a class of systems.

In addition to theoretical contributions, this work also makes key contributions to specific applications. In particular, network reconstruction algorithms are developed that extend the applicability of existing methods to general LTI systems while improving the computational complexity. Also, a passive reconstruction method was developed that enables reconstruction without actively probing the system. Finally, the structural theory developed here is used to analyze the vulnerability of a system to simultaneous attacks (coordinated or uncoordinated), enabling a novel approach to the security of cyber-physical-human systems.

Keywords: linear systems theory, dynamical structure function, structured linear fractional transformations, network semantics, system structure, network reconstruction, subsystem structure, signal structure, network vulnerability

ACKNOWLEDGMENTS

Thanks go to my beautiful wife, Leticia Suga Orikasa Chetty, my two kids, Keanu Teancum Chetty and Link Moroni Chetty, and the rest of my family, all of whom have endured with me, supported me, and cheered me throughout my time at Brigham Young University. Also, my advisor, Sean Warnick, who taught me to view the world from different perspectives, and my committee members for their help and support of my research. Finally, my colleagues at Brigham Young University – Hawaii, especially Geoff Draper for his time and effort in helping me to polish this work.

Table of Contents

List of Figures

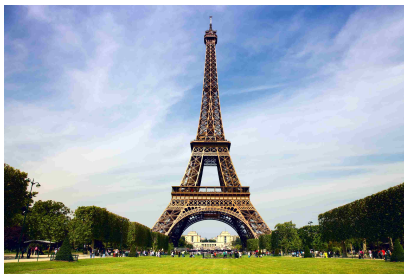
Chapter 1

Introduction

> "The lustrous, indestructible, incorruptible metal that was the unbroken surface of the planet was the foundation of the huge, metal structures that mazed the planet. They were structures connected by causeways; laced by corridors; cubbyholed by offices; basemented by the huge retail centers that covered square miles; penthoused by the glittering amusement world that sparkled into life each night."
>
> – *Isaac Asimov,* Foundation [11]

Structure is a very versatile concept that simultaneously represents the order and majesty of the Eiffel tower as well as the patterned chaos of a Picasso painting, Figure 1.1. Merriam-Webster defines structure as "something arranged in a definite pattern of organization" [6]. This interpretation of structure certainly applies to physical structures such as the Eiffel tower. The iron components of the tower are arranged in a clear, captivating pattern while still performing the unified function of ensuring the building stays erect. Similarly, the individual components in an artistic piece like Picasso's *Guernica* have a definitive pattern and organization that represents a story, concept, or idea.

Rearranging or removing components of an artistic work may cause the foundation of the work to crumble much in the same way that a building would crumble if physical components were manipulated. These examples demonstrate that structure permeates every aspect of our existence, from large interplanetary interactions millions of miles away all the way down to the atomic structure of our cells. Understanding the role of structure is critical for explaining how the world around us operates and evolves.



(a) The Eiffel tower in Paris, France [1].

(b) A painting by Pablo Picasso portraying the bombing of Guernica during the Spanish Civil War [2].

Figure 1.1: A juxtaposition of engineered structures and artistic structures.

Philosophers and scientists for millennia have attempted to explain observations of the behavior of physical phenomena. They developed tools such as mathematics and physics to detail interactions in the world around them, such as the manner in which an object moves through space or the manner in which a population evolves over time within a particular ecosystem. The resulting mathematical models demonstrate one way we interpret the world around us in order to improve our predictions of the future or our understanding of the past.

1.1 Background

This thesis looks at relationships between different mathematical models and their associated properties. In order to detail these relationships, we first define some important concepts.

1.1.1 Systems and Their Behaviors

The first concept we detail is the notion of a system. A *system* is a set of constraints on the allowed values *manifest variables* can achieve. Thus, our study of systems begins with an exploration of manifest variables. A manifest variable is a *measured* quantity that takes values from a specified set. So, for example, a traffic light takes values from the set {red, yellow, green} while my car's velocity can be represented by two real numbers–one for its speed in the "north-south" direction and the other for its "east-west" speed. Mathematically, we write w_i, $i = 1, 2, ..., n$ for a collection of n manifest variables, and the set of values each

can take can be represented as $\mathbb{W}_i$. Thus, in our example, $w_1 \in \mathbb{W}_1 = \{\text{red}, \text{yellow}, \text{green}\}$, $w_2 \in \mathbb{W}_2 = \mathbb{R}$, and $w_3 \in \mathbb{W}_3 = \mathbb{R}$. Collecting all our manifest variables together, we have

$$w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \in \{\text{red}, \text{yellow}, \text{green}\} \times \mathbb{R} \times \mathbb{R} = \mathbb{W}.$$

A *system* is then just a set of allowed values of its manifest variables, so if $w_1 =$ "red," then perhaps w_2 and w_3 must be zero; it constrains what is admissible from the entire set of possibilities defined by $\mathbb{W}$.

Dynamic systems have manifest variables that are functions of time. For our example, the traffic light w_1 may still take its values from the set $\{\text{red}, \text{yellow}, \text{green}\}$, but if it's a dynamic variable then the value it takes depends on the time $t \in \mathbb{T}$, where $\mathbb{T}$ is the set of allowed values for time. We consider both discrete time systems, where time is an integer, $t \in \mathbb{Z}$, and continuous time systems, where time is a real number $t \in \mathbb{R}$. In either case, we write $w(t) \in \mathbb{W}$ and call w a *signal* and $\mathbb{W}$ the *signal space*. The combination of allowed signals is called the *behavior* of a system, denoted $\mathcal{B} \in \mathbb{W}$. A dynamic system is defined as a triple, $(\mathbb{T}, \mathbb{W}, \mathcal{B})$.

It is important to note that a system is a set (its behavior), not an equation. Since many equations have the same solution set, we find that many equations (i.e. *models* or *representations*) characterize the same behavior, or system.

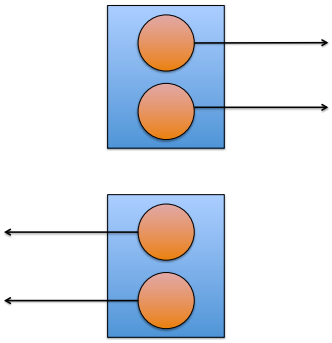
1.1.2 Interconnections of Systems

Multiple systems may be interconnected together by *variable sharing*, i.e. by equating a set of manifest variables from one system with those from another. Forcing these variables to take on the same values creates a new, composite system and the original systems become *subsystems* of the resulting composite system. The manifest variables of the composite system are a subset of the variables manifest for the set of its subsystems. This occurs because

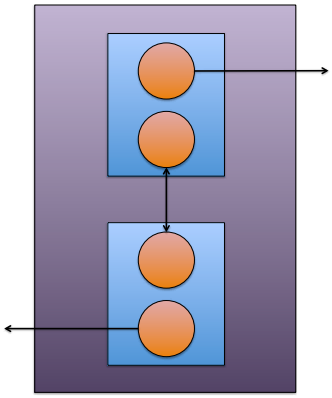
sometimes interconnections between subsystems are internal and not manifest with respect to the composite system. We call these interconnection variables *latent* or *hidden*. Note that interconnecting systems causes a distinction in what variables are manifest depending on one's *frame of reference*. Consider, for example, two systems given by $(\mathbb{T}, \mathbb{R}^2, \mathcal{B}_1)$ and $(\mathbb{T}, \mathbb{R}^2, \mathcal{B}_2)$. Each system has exactly two manifest variables, each of which can take values from the reals. If we interconnect these systems by sharing one variable from each subsystem, one possible composite system would have two manifest variables and effectively one hidden variable (since the two internal variables are forced to take on the same values due to variable sharing).

For the composite system, there are multiple notions of behavior: its manifest behavior, given by the behavior of the two manifest variables, or its complete behavior, specified by the allowed behavior of all four variables. *Black box* models are equations that accurately describe the manifest behavior of the composite system, while a complete *mechanistic* model of the system accurately represents the behavior of each subsystem as well as the interconnection structure of the composite system, thereby capturing the full behavior of all four variables of the system.

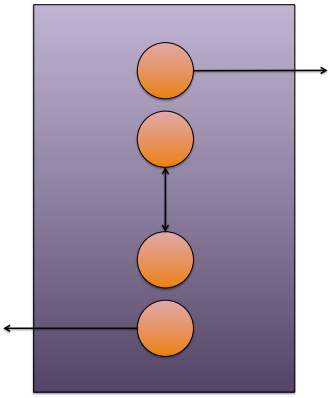
When a system is composed of interconnected subsystems, there are opportunities to not only describe its manifest behavior or its complete behavior, but also partial structure representations that capture intermediate behaviors of the system. For example, interconnecting two composite systems results in another composite system that itself has a manifest behavior and a complete behavior, but it also has an intermediate, or partial, frame of reference that only considers the manifest behavior of each subsystem. Among all such intermediate frames of reference, the one that retains the highest number of subsystems (i.e. combines the fewest subsystems into composite subsystems) while using only interconnection variables that are manifest with respect to the composite system, is called *the subsystem frame*.



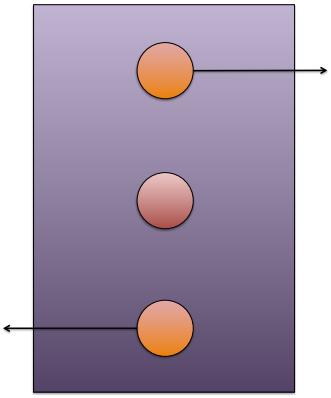
(a) Consider two systems, denoted by blue boxes, each with two manifest variables, denoted by the orange nodes. The black arrows detail information sharing, so each manifest variable shares information publicly, beyond the scope of the internal dynamics of the system.



(b) One manner of interconnecting the two systems together is through variable sharing, i.e. equating a set of manifest variables in one system with those from another. We call the corresponding interconnected system a composite system, denoted by the large purple box.



(c) The resulting composite system can be viewed from the frame of reference that it is a single system, rather than the interconnection of two systems.



(d) This perspective of the system means that the shared variable is no longer manifest, but latent or hidden. The hidden variable is denoted by the red node.

Figure 1.2: The process of interconnecting systems through variable sharing.

1.1.3 System Structures

Given a system, we see that it may have multiple frames of reference, and each frame of reference may have multiple equations, or mathematical models, describing the associated behavior. A black-box model, for example, which describes a system's manifest behavior can be associated with a graph called its *manifest structure* by identifying each manifest

variable with a node and allowing edges between nodes to indicate a dependency between variables. Similarly, a graph called the *complete computational structure* can be associated with the complete behavior of an interconnection of subsystems and a graph of the *subsystem structure* can be identified with the associated subsystem frame.

In this work, we also consider a fourth type of structure, which is the causal relationship among manifest variables without affecting other manifest variables. This structure is called the *signal structure* because it defines the structure among manifest signals, and it requires use of a directed graph to capture the inherent notion of causality built into its definition.

1.1.4 Distinctions Between Partial Structure Representations

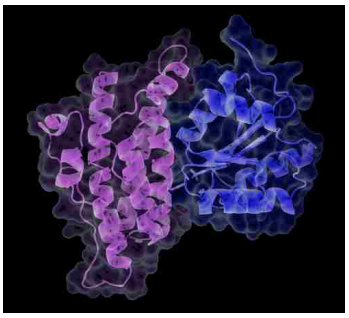
The signal structure and subsystem structure are both partial structure representations of systems and, thus, have been often confused to be the same representation of a system in the literature. In order to understand the difference between the two partial structure representations, consider a personal computer system, as in Figure 1.3a, which is made up of components such as the central processing unit, the graphics processing unit, memory, power supply, etc. Each component is distinct in that each one can be easily replaced with other, similar, compatible components without drastically changing the functionality of the overall system. Structure in this sense could be detailed mathematically using a representation like the subsystem structure. The subsystem structure, like engineered systems, are agglomerative, meaning that they both use smaller, modular components to build a larger overall system.

Some other types of systems, such as more fluidic systems (e.g. chemical processes in cells), may not have as clear a notion of subsystems as engineered systems. This means that they cannot be easily described using a representation like the subsystem structure. Consider a biochemical reaction network consisting of interacting proteins, as in Figure 1.3b. The interactions between the proteins are more difficult to describe than the interactions of an engineered system. This could be for a variety of reasons, including the fact that components may not be easily replaceable, as they are in an engineered system. Also, the manner in

which the components interact may not always be directly visible, as opposed to something like cables connecting components in a PC. In fact, connections between components in a fluidic system often must be learned through observation or experimentation. These kinds of structure could potentially be detailed mathematically using a representation like the signal structure.

Unlike the subsystem structure, the signal structure is deconstructive rather than agglomerative. The signal structure is a representation that can be learned from breaking down structure from an external perspective as opposed to constructing a system from a modular set of components.

(a) A side view of a personal computer system including a motherboard, graphics card, and power supply connected by physical wires [3].



(b) A three-dimensional X-ray crystal structure of two interacting proteins used to allow yeast cells to adapt to environmental stress [4].

Figure 1.3: Different types of systems need to be modeled with different notions of structure.

1.1.5 Why Does System Structure Matter?

There are many situations where a particular structure of a system directly impacts its dynamic behavior, and thus the need, for example, of a specific shape in a ship's hull, or a design of a complex freeway interchange, is well understood. In these cases we simply choose a structure that yields a system with the desired behavior.

Nevertheless, what about situations where very different structural choices yield exactly the same behavior, such as is frequently the case with software, electronics, and a variety of other systems? In these cases, is one structural choice preferred over others? What

criteria should one use to evaluate different structural options when the system dynamics are otherwise equivalent? Consider the following:

- **Implementation Cost**. The fact that a given manifest structure has many complete computational structures, some of which may be much more sparse than others, illustrates the important idea that the same manifest behavior of a system can often be realized from implementations with significantly different numbers of internal components. In situations where the number of components is proportional to the cost of the implementation, as is the case for many physical systems, implementation cost then becomes an important reason for understanding the structural choices available to realize a specific dynamic design.
- **Understandability**. Internal structure of a system can be important to help one understand (or hinder an outsider from understanding) how the system works. Hierarchy and modularity of subsystems are examples of methods for organizing designs so that complex systems can be more easily understood. This understandability can have a major impact on other aspects of system management, such as making the system easier to:
  - visualize,
  - promote situational awareness,
  - verify,
  - diagnose for component failure,
  - facilitate targeted access to system components, and
  - maintain.

  On the other hand, making structural choices that reduce the understandability of a system can help to secure the system from various types of infiltration, including:
  - espionage, or

– sabotage.

1.1.6 Linear Time Invariant System Representations

This thesis focuses on structures for linear time invariant system representations. For each of the four graphical structures, the associated mathematical models of linear time invariant systems are as follows:

1. The complete mechanistic model is detailed by the state space model, denoted by the matrices (A, B, C, D), (Section 2.1),
2. The black box model, or manifest structure, is detailed by the transfer function, denoted by the matrix of rational polynomials $G(s)$ where s is the Laplace variable, (Section 2.2),
3. The subsystem structure is detailed by the structured linear fractional transformation, denoted by the transfer function matrices $(Q(s), P(s))$, (Section 2.3),
4. The signal structure is detailed by the dynamical structure function, denoted by the boolean matrix N and block diagonal transfer function matrix $S(s)$, (Section 2.4).

1.2 Contributions

We now detail the specific contributions of this thesis, splitting them into two parts: theoretical contributions and advances in applications.

1.2.1 Theory

The theoretical contributions of this work are as follows:

1. **Extending the definition of the dynamical structure function to general linear time-invariant (LTI) systems**, (Section 2.4). The original definition of the dynamical structure function applied only to systems with (partial) state measurement [24]. In this work we extend the definition of the dynamical structure function to

general LTI systems, which involved a nontrivial proof of Theorem 2 ensuring that our extension is well-defined (Chapter 2). We also prove that this extended definition has several important properties, including:

(a) Invariance to block diagonal state transformations partitioned commensurate with the hidden states of the system (Chapter 2, Lemma 1),

(b) Invariance to permutations of the states (Chapter 2, Theorem 3).

2. **Generalizing necessary and sufficient conditions for identifiability of the dynamical structure function to any linear time invariant system**, (Section 3.3). The dynamical structure function is a system representation with higher structural informativity than a black box model and easier to learn from data than the complete computational structure of a system. It is represented by a pair, $(Q(s), P(s))$, of matrices of rational functions of a complex variable, and it was originally developed for the purpose of network inference [24]. The original necessary and sufficient informativity conditions for learning $Q(s)$ and $P(s)$ from data were developed for the case when $P(s)$ was diagonal. Here we generalize the conditions to any linear time invariant system.

   (a) In particular, we show that systems with non-diagonal $P(s)$ can be reconstructed, and we demonstrate the utility of this result on the reconstruction of a particular biochemical reaction network, (Section 3.4).

   (b) We also show that there exist systems that cannot have diagonal $P(s)$, so our extension is not merely a curiosity, but it is absolutely essential for some systems, (Appendix 3.6).

   (c) Finally, we demonstrate catastrophic sensitivity of our informativity conditions; being wrong about just one element (i.e. thinking $P(s)$ is diagonal when it is almost diagonal with only one non-zero off-diagonal element) results in arbitrarily bad reconstruction results, (Section 6.5.1).

3. **Developing a methodology for comparing the meaning of structures associated with different system representations and using this method to show that signal and subsystem structure can be distinct**, (Sections 4.1.1 and 5.2). Although previous work began to explore the relationships between these notions of structure [71], often they are confused in the literature. Signal structure, which is used in identification and inference algorithms, is often interpreted as the interconnection of subsystems, which is used in distributed design and multi-agent systems. Nevertheless, here we detail a precise methodology for distinguishing these concepts and reasoning about their semantics.

4. **Developing necessary and sufficient conditions for the identifiability of the subsystem structure of a class of interconnected subsystems and showing that it is always as hard or harder than reconstruction of the associated dynamical structure function**, (Section 5.3). The reason that reconstruction of the subsystem structure is harder is because it identifies a partition of the state variables (i.e. which states belong in which subsystems), while reconstruction of the signal structure remains agnostic to the presence of shared hidden state among modules in the network. This fact precisely explains the gap between network identification procedures, that identify signal structure, and decentralized control procedures, that consider interconnections of subsystems.

1.2.2 Applications

The advances in applications in this work are as follows:

1. **Adapting the existing polynomial reconstruction procedure to the non-diagonal $P(s)$ case and extending the results to improve computational complexity (from a higher order polynomial to a lower order polynomial)**, (Section 6.3). Reconstruction algorithms that estimate a system's signal structure from noisy data tend to favor highly connected structures, since more connections means more

degrees of freedom to better fit the data. As a result, one typically adds a regularization term to the objective function to penalize structures that use more connections simply to fit the noise [73]. Previous work used Akaike's information criterion to penalize overfitting and developed a polynomial search algorithm that could solve the problem [28]. The work here extends this approach to general structures, even when $P(s)$ is not diagonal, and reduces the search complexity.

2. **Developing an approach to passive reconstruction of the dynamical structure function**, (Chapter 7). Algorithms that reconstruct the dynamical structure function from data typically ensure sufficient informativity of the data by following a particular experimental procedure: to perturb each input channel with a step and observe all subsequent output channels until they reach steady state. In some applications, however, one does not have the ability to control the system inputs. As a result, *passive* reconstruction techniques are needed that simply observe the manifest signals until they are sufficiently informative to reveal the dynamical structure function and associated signal structure. To accomplish this, our work here:

   (a) Developed a time-domain representation of the DSF (Section 7.3)

   (b) Created a new time-domain reconstruction algorithm that observes measured signals and then reconstructs when they become informative enough to specify the network structure, (Section 7.2)

   (c) Detailed passive reconstruction for situations where the inputs are unmeasured e.g. stochastic inputs (Appendix 7.6)

3. **Using the partial structure modeling technology of the dynamical structure function to represent attack surfaces in complex systems and extending previous vulnerability results to situations where attackers can execute coordinated attacks against the system**, (Section 8.4). Other applications of the structural theory developed here, besides network inference, include vulnerability anal-

yses of a system. Previous work used the dynamical structure function and associated signal structure to characterize the *attack surface* of cyber-physical-human systems [45]. These models can then be analyzed to understand the size of different perturbations required to cause instability, leading to an effective denial of service attack. Our work here considers the case of coordinated attacks and demonstrates that the vulnerability calculation can effectively distinguish secure and insecure structures. Subsequent work illustrated the approach on a specific cyber-physical-human regional water management system in central Utah [25].

1.3 A Note on Related Work

Since each chapter has its own related works section, one is not provided here.

Part I

Theory: Chapters 2 - 5

Chapter 2

System Representations and the Extended Definition of the Dynamical Structure Function

(To be published in the book "Principles of Cyber-Physical Systems" as a chapter entitled "Meanings and Applications of Structure in Networks of Dynamic Systems")

This chapter describes four different mathematical representations of systems and discusses the definition and meaning of the corresponding structure for each: the generalized state space model with its complete computational structure, the transfer function and the input-output sparsity structure, structured linear fractional transformations and the subsystem structure, and the dynamical structure function with its signal structure. Each of these system representations completely characterize the dynamic behavior of a system, and thus they are equivalent from a behavioral perspective. Nevertheless, they retain varying degrees of structural information, and thus these system representations can be ordered based on their structural informativity.

In this work, a "structure" is a directed graph. We will see that different system representations specify different structural graphs, and each structural graph carries with it a unique interpretation, or meaning. We will restrict our attention to finite-dimensional, causal, deterministic linear time invariant (LTI) systems defined over continuous time, but the concepts extend naturally to the nonlinear and stochastic settings with different types of independent variables.

2.1 State Space Models and the Complete Computational Structure

State space models are the most structurally informative system representation considered here. The standard state space model for LTI systems is given by:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du, \end{aligned} \tag{2.1}$$

where $x(t) \in \mathbb{R}^n$ represents a vector of n system state variables, each defined over $t \in \mathbb{R}$; $\dot{x}(t) \in \mathbb{R}^n$ represents the time derivative of these state variables; $u(t) \in \mathbb{R}^m$ are controlled inputs into the system; and $y(t) \in \mathbb{R}^p$ are measured outputs. Recall that states have a particular meaning, being the information necessary to characterize the future evolution of the system. That is to say, given the values of x at some time (which will be labeled $t = 0$ without loss of generality), only future values of the input, $u(t)$ for $t \geq 0$, are needed to completely specify the evolution of the system for all times $t \geq 0$. This representation is sufficiently detailed to completely characterize both the transfer function and the dynamical structure function of a system, with their corresponding structures.

Nevertheless, this standard state space model does not differentiate between systems with different subsystem structures. For example, consider two systems in feedback. One can easily compute the closed-loop dynamics of such an interconnection and represent them with a single standard state space model. Nevertheless, if presented with this closed-loop model, one can not determine what the two subsystems are that generate it. This failure to distinguish different subsystem structures comes from the standard state space model's lack of representation power to distinguish between the composition of functions (see Example 1).

To distinguish different subsystem structures, we need to differentiate between behaviorally equivalent computations such as 1) $f(x) = x$, 2) $f(x) = 2(0.5x)$ and 3) $f(x) = 0.3x + 0.7x$. We accomplish this by introducing *auxiliary variables*, w, that represent intermediate stages of computation. In this way we can differentiate 1) $f(x) = x$ from 2)

$f(x) = 2w$ and $w = 0.5x$ or 3) $f(x) = w_1 + w_2$ and $w_1 = 0.3x$ and $w_2 = 0.7x$, since each of these different ways of computing the same functional relationship involve zero, one, or two auxiliary variables, respectively. The auxiliary variables that are specified, say, in a system's "blueprint" or manifest directly to observers, help us distinguish the system's actual computational structure from others we could imagine.

Introducing auxiliary variables into the standard state space model characterizes a differential-algebraic system of equations capable of uniquely specifying all three of the other system representations discussed here. We call this modified system of equations the *generalized state space model* of a system, and represent it as

$$\begin{aligned} \dot{x} &= Ax + \hat{A}w + Bu \\ w &= \bar{A}x + \tilde{A}w + \bar{B}u \\ y &= Cx + \bar{C}w + Du \end{aligned} \tag{2.2}$$

where $w \in \mathbb{R}^l$, $\hat{A} \in \mathbb{R}^{n \times l}$, $\bar{A} \in \mathbb{R}^{l \times n}$, $\tilde{A} \in \mathbb{R}^{l \times l}$, $\bar{B} \in \mathbb{R}^{l \times m}$, and $\bar{C} \in \mathbb{R}^{p \times l}$. The number of auxiliary variables, l, is called the *intricacy* of the generalized state space model. Choosing $\tilde{A}$ so that $I - \tilde{A}$ is invertible yields a differentiability index of zero. This ensures that the auxiliary variables can always be algebraically eliminated from the system, producing a dynamically equivalent standard state space model (2.1). We call this equivalent standard state space model the *zero-intricacy* realization or representation of a given generalized state space model (2.2).

Example 1. *Consider the feedback interconnection of two systems, given by*

$$\begin{aligned} \dot{x}_1 &= A_1 x_1 + B_1 r_1 & \qquad \dot{x}_2 &= A_2 x_2 + B_2 r_2 \\ y_1 &= C_1 x_1 & \qquad y_2 &= C_2 x_2 \end{aligned}$$

with $r_1 = u_1 + y_2$ and $r_2 = u_2 + y_1$, where u_1 and u_2 are exogenous inputs to the closed-loop system, and y_1 and y_2 are measured outputs from the closed-loop system. Defining $w_1 = y_1$ and

$w_2 = y_2$, *we obtain the following generalized state space model of the feedback interconnection:*

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & B_1 \\ B_2 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} + \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{2.3}$$

Note that $I - \tilde{A}$ is invertible, thus enabling us to easily eliminate w from the equations. Doing so yields the zero-intricacy representation of the feedback interconnection:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_1 & B_1C_2 \\ B_2C_1 & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{2.4}$$

Although these representations are dynamically equivalent, meaning that (2.3) and (2.4) generate identical state and output trajectories if they are given the same initial condition x_o and input trajectory $u(t)$, (2.3) encodes information to uniquely specify the original subsystems and their feedback interconnection structure, while (2.4) does not.

Example 1 illustrates a generalized state space model and the corresponding zero-intricacy realization of a system composed of the interconnection of multiple subsystems. In fact, whenever $I - \tilde{A}$ is invertible, every generalized state space model has a unique,

well-defined zero-intricacy realization. Likewise, every zero-intricacy state space model is dynamically equivalent to a rich variety of generalized state space models of any positive intricacy; these generalized state space models differ only in how their computations are performed, or in their underlying computational structure. We call this structure of the most refined generalized state space description of a system, even zero-intricacy ones, the *complete computational structure*, and all other notions of system structure discussed in this work can be derived directly from it.

Definition 1 (Complete Computational Structure)**.** *Given a generalized state space model, as in (2.2), its* complete computational structure *is a weighted directed graph $\mathscr{C}$ with vertex set $V(\mathscr{C})$ and edge set $E(\mathscr{C})$ given by:*

- $V(\mathscr{C}) = \{u_1, ..., u_m, x_1, ..., x_n, w_1, ..., w_l, y_1, ..., y_p\}$*, and*
- $E(\mathscr{C})$ *is specified by the nonzero entries of the adjacency matrix $\mathscr{A}(\mathscr{C})$, where*

$$\mathscr{A}(\mathscr{C}) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ B & A & \hat{A} & 0 \\ \bar{B} & \bar{A} & \tilde{A} & 0 \\ D & C & \bar{C} & 0 \end{bmatrix}^T . \tag{2.5}$$

That is to say, a potential edge from $v_i \in V(\mathscr{C})$ to $v_j \in V(\mathscr{C})$ has weight $\mathscr{A}(\mathscr{C})_{ij}$, but we only recognize the existence of edges with non-zero weight.

The generalized state space model (2.2) encodes information about how the system performs the computations necessary to realize its dynamic behavior. It is like an *information blueprint* of how specific components are interconnected to access information from input signals; how this information is represented (in a specific coordinate system) and combined with other data retrieved from memory; how these new calculations are stored; and how all of this data combines to produce measurable output signals. The meaning, then, of the complete computational structure characterized by (2.5), is the *information architecture* of a very

specific computation system: how information is represented, transformed, and flows through the system. Note that there is a distinction between "physical structure" and state space models; in some cases, the particular basis specified by a state space model is more detailed than the physical structure may suggest. For example, consider an inertial mass. This mass behaves like a second order system according to Newton's Second Law of Motion, but it is not clear whether states of the system are necessarily position and velocity, or whether they are some linear combinations of position and velocity. Exactly how some systems represent and store information may be unclear, but if it were known, state space models are capable of representing this refined level of structural knowledge. These models (the generalized state space model and its associated complete computational structure) then become the most refined knowledge of our system, ground truth from which all other representations can be compared.

Note that because intricacy variables can always be eliminated from a generalized state description without changing its dynamics, the most refined generalized state space model, with intricacy $l > 0$, immediately defines a particular sequence of state space models indexed by their intricacies, $l-1, ..., 0$. Each of these coarser models has a structure associated with it that we call a *computational structure*, but we reserve the descriptor, *complete computational structure* for the most refined structural specification of the system; once the complete computational structure is specified, even if it has zero-intricacy, all other hypothetical refinements are considered fictitious while any agglomerative structure derived from it is a valid notion of structure for the system.

Example 2. *Making Example 1 concrete, consider the following two systems:*

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}-1 & 2\\ 0 & -2\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}2 & -1\\ -1 & 1\end{bmatrix}\begin{bmatrix}r_1\\ r_2\end{bmatrix} \qquad \begin{bmatrix}\dot{x}_3\\ \dot{x}_4\\ \dot{x}_5\end{bmatrix} = \begin{bmatrix}-5 & -4 & 2\\ 3 & 2 & -1\\ 0 & 0 & -3\end{bmatrix}\begin{bmatrix}x_3\\ x_4\\ x_5\end{bmatrix} + \begin{bmatrix}0 & -1\\ 0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}r_3\\ r_4\end{bmatrix}$$

$$\begin{bmatrix}y_1\\ y_2\end{bmatrix} = \begin{bmatrix}1 & 2\\ 1 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}, \qquad \begin{bmatrix}y_3\\ y_4\end{bmatrix} = \begin{bmatrix}1 & 2 & 0\\ 1 & 1 & 0\end{bmatrix}\begin{bmatrix}x_3\\ x_4\\ x_5\end{bmatrix},$$

interconnected in feedback, so that

$$\begin{bmatrix}r_1\\ r_2\end{bmatrix} = \begin{bmatrix}y_3\\ y_4\end{bmatrix} + \begin{bmatrix}u_1\\ u_2\end{bmatrix}, \qquad \begin{bmatrix}r_3\\ r_4\end{bmatrix} = \begin{bmatrix}y_1\\ y_2\end{bmatrix} + \begin{bmatrix}u_3\\ u_4\end{bmatrix},$$

leading to the following generalized state space model:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\\ \dot{x}_5\end{bmatrix} = \begin{bmatrix}-1 & 2 & 0 & 0 & 0\\ 0 & -2 & 0 & 0 & 0\\ 0 & 0 & -5 & -4 & 2\\ 0 & 0 & 3 & 2 & -1\\ 0 & 0 & 0 & 0 & -3\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\\ x_5\end{bmatrix} + \begin{bmatrix}0 & 0 & 2 & -1\\ 0 & 0 & -1 & 1\\ 0 & -1 & 0 & 0\\ 0 & 1 & 0 & 0\\ 1 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}w_1\\ w_2\\ w_3\\ w_4\end{bmatrix} + \begin{bmatrix}2 & -1 & 0 & 0\\ -1 & 1 & 0 & 0\\ 0 & 0 & 0 & -1\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}u_1\\ u_2\\ u_3\\ u_4\end{bmatrix}$$

$$\begin{bmatrix}w_1\\ w_2\\ w_3\\ w_4\end{bmatrix} = \begin{bmatrix}1 & 2 & 0 & 0 & 0\\ 1 & 1 & 0 & 0 & 0\\ 0 & 0 & 1 & 2 & 0\\ 0 & 0 & 1 & 1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\\ x_5\end{bmatrix} + \begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}w_1\\ w_2\\ w_3\\ w_4\end{bmatrix} + \begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}u_1\\ u_2\\ u_3\\ u_4\end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}. \quad (2.6)$$

The complete computational structure of this system, given in (2.6), is shown in Figure 2.1.

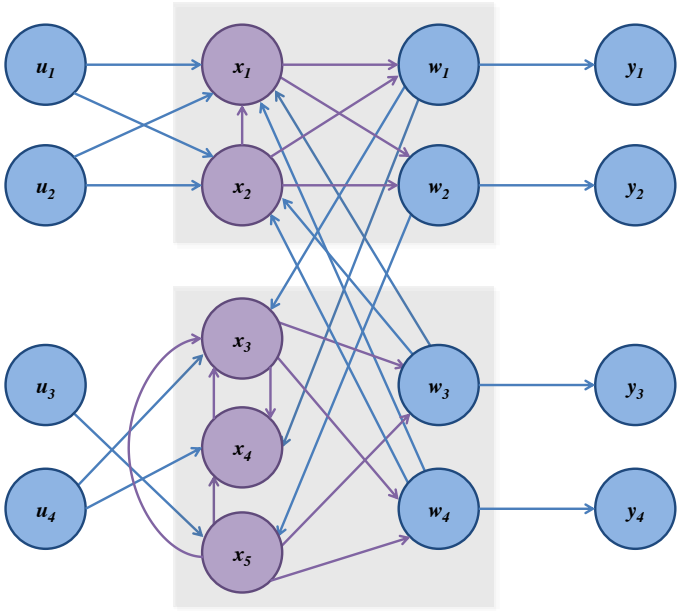


Figure 2.1: Complete computational structure of the generalized state space model from (2.6). Blue nodes are manifest variables, while purple nodes indicate hidden variables. Notice that the original feedback structure of subsystems, reflected by gray boxes, is preserved, since the only interaction between subsystems is through manifest variables.

The zero-intricacy realization of this generalized state space model, (2.6), is then given by:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = \begin{bmatrix} -1 & 2 & 1 & 3 & 0 \\ 0 & -2 & 0 & -1 & 0 \\ -1 & -1 & -5 & -4 & 2 \\ 1 & 1 & 3 & 2 & -1 \\ 1 & 2 & 0 & 0 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} + \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \tag{2.7}$$

The computational structure of the zero-intricacy realization, given in (2.7), is shown in Figure 2.2. Notice the differences with the complete computational structure shown in Figure 2.1. For example, the complete computational structure has nodes for auxiliary variables, w, while the computational structure of the zero-intricacy realization does not. Also, original subsystem structure is preserved in the complete computational structure, highlighted by the background gray boxes, while it is lost in the computational structure of the zero-intricacy realization, resulting in no distinguishable subsystems.

2.2 Functional System Descriptions and the Manifest Structure

While state space models are the most structurally informative system representations, functional system descriptions, such as convolution models or transfer functions, are at the other end of the spectrum. These "black box" representations of a system are capable of describing the same dynamic behavior as their state space counterparts, yet they do not

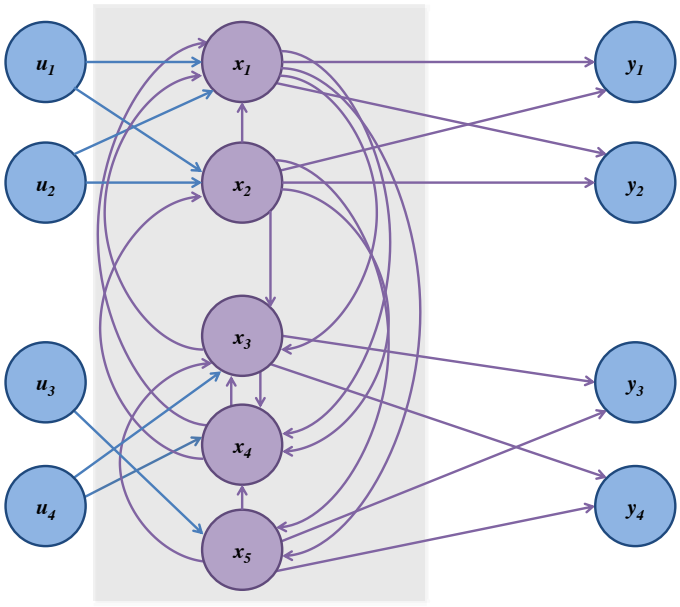


Figure 2.2: Computational structure of the zero-intricacy realization (2.7) of the generalized state space model in (2.6). Like Figure 2.1, blue nodes indicate manifest variables while purple nodes are hidden variables. Notice that the original subsystem structure is lost, and only a single subsystem remains visible from manifest variables.

model the detailed interactions among system components the way state space representations do[1].

This inability to convey detailed structural information is not necessarily a weakness, however. For example, functional representations need fewer parameters to characterize a given dynamic behavior, making them easier to learn from data (called *system identification* [31, 57]) than their state space counterparts. Moreover, their parsimonious description of a system's dynamics creates an important distinction between a system's behavior and how it realizes that behavior, enabling a concerted focus on the design of a system's dynamics without worrying about implementation.

Just as high-level programming languages abstract many of the details of the computer they run on, functional system descriptions are high-level abstractions of state space models.

[1]Although transfer functions and convolution models of LTI systems assume zero initial conditions, the impact of a non-zero initial condition is easily modeled with the addition of an appropriately designed external disturbance.

In particular, the specific processes a state realization uses to decide which information is stored in which parts of the state vector correspond to memory management activities that are completely invisible to a functional description of a system. This distinction is further exemplified by noting that state space models are imperative descriptions of a system, encoding computations in terms of the time evolution of the system state, while functional descriptions are inherently declarative, specifying what the system does without prescribing how it should do it.

The result of this high-level/low-level relationship between functional system descriptions and state space models is a one-to-many relationship between the two model classes. That is, every state space model has a zero-intricacy realization as in Equation (2.1) that identifies a unique functional system description, whether it be the impulse response matrix of a convolution model or a transfer function matrix, given by:

$$\begin{aligned} y(t) &= h(t) * u(t) & \qquad Y(s) &= H(s)U(s) \\ h(t) &= Ce^{At}B + D\delta(t), & \qquad H(s) &= C(SI - A)^{-1}B + D, \end{aligned} \tag{2.8}$$

where $*$ denotes convolution, $\delta(t)$ is the Dirac delta function, $h(t)$ is the system's $p \times m$ impulse response matrix, $Y(s)$ and $U(s)$ are the Laplace transforms of $y(t)$ and $u(t)$, and $H(s)$ is the system's $p \times m$ transfer function matrix–which is also the Laplace transform of $h(t)$.

Note, however, that there are many state space models that specify the same impulse response or transfer function; each of these state space models specifies a different implementation (or realization) of the same dynamic behavior. Among all these state realizations of a given functional description of a system, some have fewer states than others. In fact, systems with functional descriptions that can be described by finite-dimensional LTI state space models[2] have a unique integer, n, associated with them called the *Smith-McMillan degree*.

[2]Although all LTI state space models have transfer functions, not all transfer functions have state space realizations. This is because the imperative nature of state space models demand that they are *causal*, meaning that future values of manifest variables only depend on past and present values of manifest variables.

This degree is the minimal number of states necessary for any state space realization of the system. Nevertheless, even restricting attention to state space models with order equal to the Smith-McMillan degree does not yield a unique state realization; given a minimal realization (A, B, C, D) of a transfer function $H(s)$, any $n \times n$ transformation, T, yields another minimal realization $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$ given by:

$$\hat{A} = TAT^{-1}, \quad \hat{B} = TB, \quad \hat{C} = CT^{-1}, \quad \hat{D} = D \tag{2.9}$$

such that $C(sI - A)^{-1}B + D = H(s) = \hat{C}(sI - \hat{A})^{-1}\hat{B} + \hat{D}$. Thus, even among minimal realizations, there are infinitely many implementations of a given dynamic behavior specified by a functional description such as $H(s)$, and these implementations differ only in their structural properties.

The functional description of a system, however, retains only the structural properties that are common among all of its state realizations, which is precisely the mathematical structure of the functional description itself. This structure describes the internal closed-loop relationships among manifest variables, and therefore is called the *manifest structure*.

Definition 2 (Manifest Structure)**.** *Given a generalized state space model, as in (2.2), identified by a functional system description, as in (2.8), its* manifest structure *is a weighted directed graph $\mathscr{M}$ with vertex set $V(\mathscr{M})$ and edge set $E(\mathscr{M})$ given by:*

- *$V(\mathscr{M}) = \{u_1, ..., u_m, y_1, ..., y_p\}$, each representing a manifest signal of the system, and*
- *$E(\mathscr{M})$ has an edge from u_i to y_j, labeled by either H_{ji} or h_{ji}, provided they are non-zero.*

Note that when a system's manifest variables partition naturally into inputs and outputs, then its manifest structure is a bipartite graph, with directed edges from inputs to outputs.

An alternative definition of the manifest structure characterizes $\mathscr{M}$ directly from $\mathscr{C}$ using only graphical properties (which is useful when extending these results to the nonlinear

Transfer functions that are *proper* rational functions of the Laplace variable correspond to causal finite dimensional LTI systems; we do not concern ourselves with other kinds in this work.

setting). In that case, we say $\mathscr{M}$ has an edge from u_i to y_j if the net impact of all paths in $\mathscr{C}$ from u_i to y_j is non-zero, or, equivalently, if every equivalent realization of the system, specified by a transformation T as in (2.9), with complete computational structure $\mathscr{C}_T$, has a path from u_i to y_j.

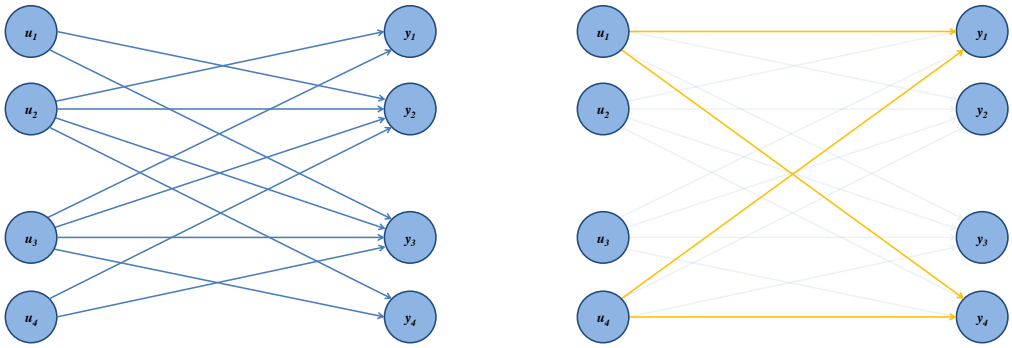
Example 3. *Consider the zero-intricacy state space model in (2.7) from Example 2. The corresponding transfer function is given by:*

$$H(s) = C(sI - A)^{-1}B + D =$$

$$\begin{bmatrix} 0 & \frac{s^2+5s+6}{s^3+6s^2+11s+5} & \frac{1}{s^3+6s^2+11s+5} & 0 \\ \frac{s+1}{s^2+3s+1} & \frac{s^3+6s^2+11s+8}{s^5+9s^4+30s^3+44s^2+26s+5} & \frac{3s+3}{s^5+9s^4+30s^3+44s^2+26s+5} & \frac{1}{s^2+3s+1} \\ \frac{1}{s^2+3s+1} & \frac{s^2+7s+10}{s^5+9s^4+30s^3+44s^2+26s+5} & \frac{2s^2+6s+5}{s^5+9s^4+30s^3+44s^2+26s+5} & \frac{s+2}{s^2+3s+1} \\ 0 & \frac{1}{s^3+6s^2+11s+5} & \frac{s+1}{s^3+6s^2+11s+5} & 0 \end{bmatrix} \tag{2.10}$$

The manifest structure corresponding to this transfer function, (2.10), that represents the internal closed-loop pathways from inputs to outputs of the system in (2.2) is given in Figure 2.3.

Note that in some cases, although a pathway exists from an input to an output in the system's complete computational structure, it is possible that the corresponding transfer function from the input to the output is zero. For example, notice that although paths exist from every input to every output in the computational structure of the zero-intricacy realization generating H (Figure 2.2), H_{11}, H_{41}, H_{14}, *and* H_{44} *are nevertheless all zero. Thus, we see that the existence of paths from* u_i *to* y_j *is not sufficient for* H_{ij} *to be nonzero; the closed-loop, net effect of* all *paths from* u_i *to* y_j *must be nonzero for* H_{ij} *to be nonzero; exact cancellations, which can be common in software and other engineered systems, can generate zeros in the functional description.*



(a) Manifest structure of the system with transfer function (2.10)

(b) Missing edges in the manifest structure, corresponding to zero elements in H.

Figure 2.3: Manifest structure of the same system from Figures 2.1 and 2.2. Notice the lack of edges from u_1 to y_1 and y_4, and from u_4 to y_1 and y_4, corresponding to associated zeros in $H(s)$. These missing links are highlighted in Figure 2.3b. Note that these links are missing in the manifest structure even though paths exist in Figure 2.2 from every input to every output.

2.3 Structured Linear Fractional Transformations and the Subsystem Structure

Having identified the complete computational structure as the most informative structural representation, and the manifest structure as the least, we now explore the most common intermediate structural representation: the interconnection of subsystems. Subsystem structure is less informative than the complete computational structure because it does not reveal the internal structure of subsystems. On the other hand, subsystem structure can be more informative than manifest structure because it reveals the interconnection pattern among subsystems.

To isolate and represent the interconnection pattern of subsystems for a given system, begin by considering a set of q subsystems, $S = \{ \; S_1 \quad S_2 \quad \ldots \quad S_q \; \}$, interconnected into a composite system, H. It is conceivable that each of these subsystems are themselves divisible into constituent subsystems, or that not all of the q subsystems are discernible from H's manifest variables, so we specify the level of modeling abstraction by:

1. Modeling each of the q constituent subsystems with a suitable functional description, such as a proper or strictly proper transfer function $S_i(s)$, $i = 1, 2, ..., q$, or a *single-subsystem* state space realization, characterized as a generalized state space model with subsystem structure consisting of a single subsystem, so that no further division of the subsystems is possible, and

2. Ensuring that each of the subsystem's outputs, w_i, is a measured output of the composite system H, so $y = [\ w_1^T \quad w_2^T \quad ... \quad w_q^T\ ]^T$, where y is the output of H.

Note that each subsystem is *distinct*, meaning that state variables internal to one subsystem are different from those of the other subsystems, yielding no mechanism for interaction except through their respective manifest variables. Let u be a vector of external inputs; v_i and w_i be the vectors of inputs and outputs for system S_i; and v and w be the stacked inputs and outputs from all systems, $v = [\ v_1^T \quad v_2^T \quad ... \quad v_q^T\ ]^T$ and $w = [\ w_1^T \quad w_2^T \quad ... \quad w_q^T\ ]^T$, so that $w = y$. Interconnecting these systems then means defining binary matrices L and K such that:

$$\begin{bmatrix} L & K \end{bmatrix} \begin{bmatrix} u \\ w \end{bmatrix} = v. \tag{2.11}$$

Our convention is that the process of interconnection only allows the selection of particular signals and possibly adding them together, thus restricting the interconnection matrices, L and K, to have elements with values of either zero or one; all other computations are part of the systems in S. Further, we assume that the resulting interconnection is *well-posed*, meaning that all signals within H are uniquely specified for any value of external inputs and underlying state variables [75]. This assumption ensures that the proposed interconnection is physically sensible and not merely a mathematical artifact.

The composite system, H, is then clearly defined by the structured linear fractional transformation (LFT) as in Figure 2.4, given by:

$$N \begin{bmatrix} u \\ w \end{bmatrix} = \begin{bmatrix} y \\ v \end{bmatrix}, \qquad (2.12)$$
$$w = Sv,$$

where

$$N = \begin{bmatrix} 0 & I \\ L & K \end{bmatrix}, \qquad S = \begin{bmatrix} S_1 & 0 & \dots & 0 \\ 0 & S_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & & \dots & S_q \end{bmatrix} \qquad (2.13)$$

and S_i can be represented by either a suitable functional description, such as a proper or strictly proper transfer function matrix or the associated impulse response matrix of a convolution model, or by any single-subsystem generalized state realization. The symbol S is overloaded, representing both the set of subsystems and the decoupled operator of subsystem models in (2.13), but the appropriate meaning should always be clear from context. Equations (2.12) and (2.13) characterize H as a structured LFT in terms of S. Combining these equations yields, for example, $Y(s) = [S(s)(I - KS(s))^{-1}L]U(s)$, implying that $H(s) = S(s)(I - KS(s))^{-1}L$, where $Y(s)$ and $U(s)$ are the Laplace transforms of $y(t)$ and $u(t)$, respectively; similarly, a well-specified expression can be obtained for h directly in the time domain. The functional description of the composite system, H, in either the time or frequency domain, is completely specified by the structured LFT description in (2.12) and (2.13).

Although the structured LFT completely specifies the functional description of the composite system, H, the structured LFT does not have enough structural information to specify H's complete computational structure or its associated generalized state space description. To do so, it would need information about the "true" structure of each constituent

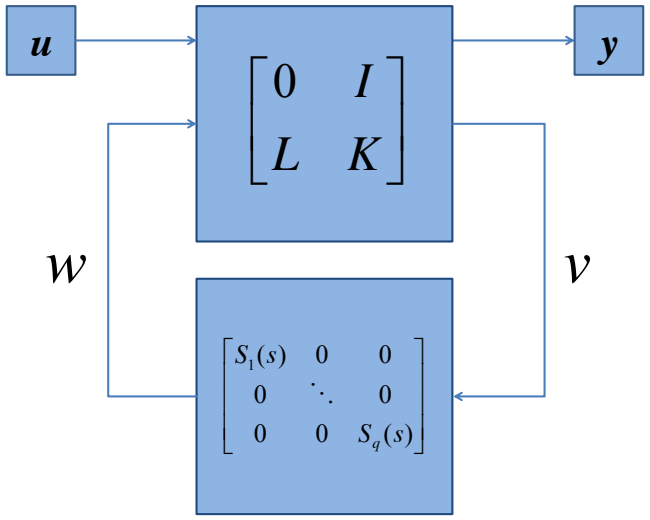


Figure 2.4: A structured linear fractional transformation revealing the interconnection structure among subsystems in binary matrices L and K.

subsystem. This point may be clear when S is specified by a functional description for each subsystem, such as its transfer function, but it becomes more subtle when S is specified by a generalized state space model for each subsystem. In this case, it is important to understand that the state space model for each subsystem in S can be *any* single-subsystem realization of the associated transfer function, $S_i(s)$, since the structured LFT does not use any information about the internal structure of its subsystems. To realize the "true" generalized state description of H an its associated complete computational structure, one must have accurate descriptions of the complete computational structures for each constituent subsystem to complement the "interconnection" information in the structured LFT.

The structured LFT reveals the interconnection structure among subsystems, encoded in the binary *interconnection matrix*, N, in general, and in L and K in particular. Note that the interconnection structure in N is unaffected by whether the subsystems in S are represented by state space models or transfer functions. The internal computational structure of subsystems, revealed by state models of subsystems but not transfer function representations of subsystems, is not used when representing the subsystem structure of

a system–only the interconnection structure *among* subsystems, not *within* subsystems, is relevant for this representation.

Aggregating L and K appropriately to account for the potentially multi-input multi-output nature of the constituent subsystems yields adjacency matrices from which the composite system's subsystem structure can be built. To accomplish this, let e_{v_i} denote the vector of ones with length equal to the length of vector v_i. We then define the aggregation matrices

$$A_v = \begin{bmatrix} e_{v_1}^T & 0 & ... & 0 \\ 0 & e_{v_2}^T & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & ... & e_{v_q}^T \end{bmatrix}, \qquad A_w = \begin{bmatrix} e_{w_1}^T & 0 & ... & 0 \\ 0 & e_{w_2}^T & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & ... & e_{w_q}^T \end{bmatrix}, \tag{2.14}$$

and use them to create the adjacency matrices:

$$\mathscr{A}(L) = sgn(A_v L)^T, \qquad \mathscr{A}(K) = sgn(A_v K A_w^T)^T, \tag{2.15}$$

where $sgn(\cdot)$ denotes the sign function, yielding a value of one for positive entries, zero for zero, and negative one for negative entries (which can never occur in this case). With these definitions, we are now prepared to characterize a system's *subsystem structure*:

Definition 3 (Subsystem Structure)**.** *Given a generalized state space model, as in (2.2), identified by a structured LFT, (N, S), as in (2.12) and (2.13) and with associated aggregation matrices as in (2.14) and adjacency matrices as in (2.15) , its* subsystem structure *is a weighted directed graph $\mathscr{S}$ with vertex set $V(\mathscr{S})$ and edge set $E(\mathscr{S})$ given by:*

- $V(\mathscr{S}) = \{u_1, ..., u_m, S_1, ..., S_q, y_1, ..., y_p\}$*, representing input signals, subsystems, and output signals, respectively.*
- $E(\mathscr{S})$ *has an edge from*
  - u_i *to* S_j *if* $\mathscr{A}(L)_{ij} = 1$*, labeled* u_i*;*

– S_i *to* S_j *if* $\mathscr{A}(K)_{ij} = 1$, *labeled* w_i;

– S_i *to* y_j *if* $(A_w)_{ij} = 1$, *labeled* y_j.

Note that the subsystem structure is qualitatively different from either the complete computational structure or the manifest structure in a few ways. First, while all the nodes of either the complete computational structure or the manifest structure represent *signals*, the nodes of the subsystem structure represent *systems*, namely the subsystems and exosystems associated with the generation of each input or measurement of each output signal. As a result, we often denote the nodes in the subsystem structure with a different shape, e.g. rectangles instead of circles, to highlight this distinction (see Figure 2.5e). Also, the edges in both the complete computational structure and the manifest structure are labeled to represent *systems*, while the edges in the subsystem structure are labeled with the names of *signals*. These distinctions make it clear that the subsystem structure carries the interpretation of a *block diagram*, while the other structures are *signal flow graphs*.

The definition of subsystem structure given above characterizes the graph in terms N and S. Nevertheless, the subsystem structure can be obtained directly from the complete computational structure, which not only lends a graphical interpretation to the concept of a subsystem, but naturally facilitates the extension of the definitions to the nonlinear and stochastic setting. We achieve this by first extending the definition of a *manifest node* or *manifest signal* of $\mathscr{C}$ to include any node representing a signal identically equal to a manifest signal, u_i or y_j. We then consider the subgraph of $\mathscr{C}$ obtained by 1) removing all input nodes and any outgoing edges leaving them, 2) removing all output nodes and any incoming edges entering them, and 3) removing all outgoing edges leaving any remaining manifest nodes. This subgraph, $\mathscr{H}$ is the *hidden structure* of $\mathscr{C}$, and it immediately reveals its subsystems and their interconnection, as follows:

Theorem 1. *Consider a system H characterized by a structured LFT, (N, S). Construct a complete computational structure for H, as in (2.2), by realizing each subsystem in S*

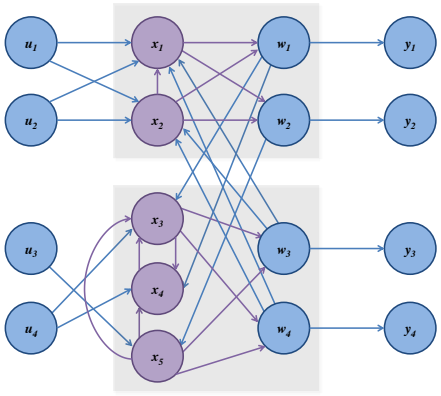
with a single-subsystem state space model, and let $\mathscr{C}$ be the resulting complete computational structure. Then every connected component of $\mathscr{H}$, the hidden structure of $\mathscr{C}$, corresponds to a distinct subsystem in S.

Proof. Since each subsystem is realized by a single-subsystem state space realization, variables internal to each subsystem correspond to nodes of $\mathscr{C}$ that are connected to each other. Moreover, since all outputs of S are manifest, and S is diagonal, these connected components can only be interconnected by manifest signals. By removing all outgoing edges from internal manifest nodes in $\mathscr{C}$, as well as removing all input and output nodes and their associated edges, $\mathscr{H}$ isolates each subsystem so the remaining connected components of $\mathscr{H}$ correspond to the subsystems in S. □

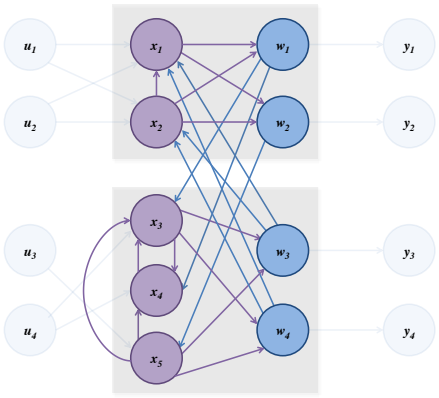
The next example illustrates this procedure of obtaining a system's subsystem structure directly from its complete computational structure.

Example 4. *Consider the generalized state space model of two subsystems in feedback from Example 2, given by Equation (2.6). Figures 2.1 and 2.5a illustrate the system's complete computational structure, $\mathscr{C}$, and we can generate its subsystem structure by identifying the connected components in the hidden structure of $\mathscr{C}$, as demonstrated in Figure 2.5:*

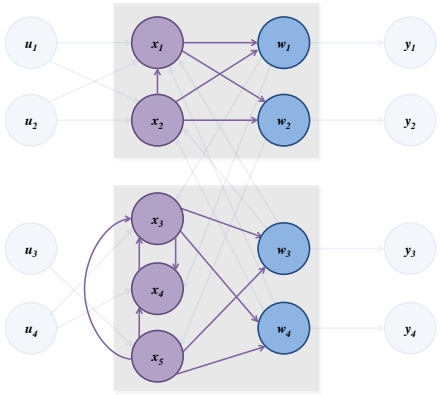
The process of constructing a system's subsystem structure from its complete computational structure involves 1) identifying all manifest nodes in $\mathscr{C}$, 2) removing all input and output nodes and their adjacent edges, 3) removing all outgoing edges from any remaining manifest nodes. These three steps construct the hidden structure, $\mathscr{H}$, and each connected component in $\mathscr{H}$ corresponds to a subsystem. Compress these connected components into single subsystem nodes and replace the input and output nodes as exosystems (instead of signals). Replace all removed edges following the convention that if a node in $\mathscr{C}$ is no longer in $\mathscr{S}$, connect the edge to the corresponding subsystem node. This may lead to multiple edges between nodes in $\mathscr{S}$ (e.g. between subsystems), so we compress these edges into a single edge and change the label to be a vector label, reflecting the multiple signals on that edge.



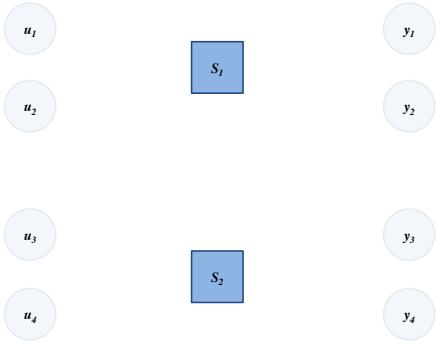
(a) Step 1: Identify the manifest variables (shaded blue).



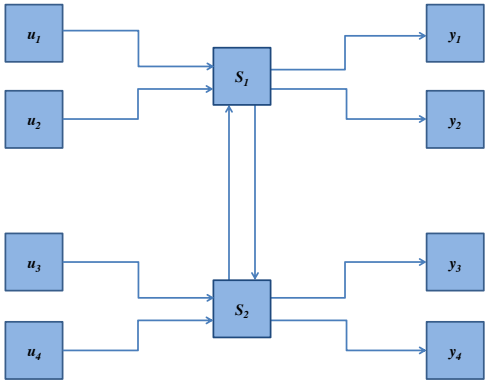
(b) Step 2: Remove all input and output nodes, along with any edges adjacent to these nodes.



(c) Step 3: Remove any outgoing edges from any remaining manifest variables.



(d) Step 4: Remaining connected components correspond to subsystems.



(e) Step 5: Reintroduce the input and output variables as exosystem nodes, and replace all removed edges, compressing any duplicate edges into a single edge.

Figure 2.5: Subsystem structure, built from a system's complete computational structure.

Now, compare the resulting subsystem structure with the results we obtain if we work directly from the equations defining the original subsystems in Example 2 leading up to Equation (2.6). If we find the transfer function of each subsystem individually, build the associated subsystem matrix S, and then interconnect appropriately, we recover the following structured LFT:

$$N = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{2.16}$$

$$S = \frac{1}{s^2+3s+2} \left[\begin{array}{cccc|cccc} 0 & s+2 & 0 & s+2 & 0 & 0 & 0 & 0 \\ s+1 & 1 & s+1 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & \frac{2}{s+3} & s+2 & \frac{2}{s+3} & s+2 \\ 0 & 0 & 0 & 0 & \frac{s+1}{s+3} & 0 & \frac{s+1}{s+3} & 0 \end{array}\right]$$

Compare the results of the structured LFT with the signal structure in Figure 2.5e. Notice that building the subsystem structure according to Definition 3 leads to the same result; both processes construct the same graph. Nevertheless, building subsystem structure directly from $\mathscr{C}$ sheds insight into the meaning of subsystems, as the connected components of the hidden structure of $\mathscr{C}$.

These procedures uniquely specify (N, S) and $\mathscr{S}$ from a generalized state space model and its complete computational structure, $\mathscr{C}$. This implies that the system models and their associated structural representations considered so far produce a totally ordered set with respect to the relation, "uniquely specified by." These are, in order of increasing structural informativity:

1. Functional system descriptions and the manifest structure, (uniquely specified by)
2. Structured LFTs and the subsystem structure, (uniquely specified by)
3. Generalized state space models and the complete computational structure.

The next section considers an alternative approach for representing systems, focusing on the interaction among manifest signals as opposed to the interconnection among subsystems.

2.4 Dynamical Structure Functions and the Signal Structure

One of the difficulties in learning a system's subsystem structure from data is that it necessarily partitions the system states into subsystem groups, so one must be able to identify the correct subsystem for each state variable–even those that are "hidden," or not directly manifest. This section considers a system representation that precisely characterizes the interaction between manifest signals without drawing any conclusions about "hidden" variables. This ability to remain agnostic about the structural role of hidden variables not only makes this representation easier to learn from data, but it also makes it extremely useful for describing systems with a "fluidic" component that makes the very idea of subsystems difficult to conceptualize, such as chemical reaction processes or market behavior.

This representation, called the *dynamical structure function* (DSF), like the structured LFT and the subsystem structure, is part of a totally ordered set with respect to the relation, "uniquely specified by." This is, in order of increasing structural informativity:

1. Functional system descriptions and the manifest structure, (uniquely specified by)
2. Dynamical structure functions and the signal structure, (uniquely specified by)

3. Generalized state space models and the complete computational structure.

Note that representations of the signal structure similar to the dynamical structure function are found across the literature, especially in the area of network reconstruction, [18, 36, 42]. Here we define a system's dynamical structure function by considering the zero-intricacy realization of each subsystem in a generalized state space model, as in (2.1). The overall dynamical structure function of a system is determined by finding the dynamical structure function of each subsystem and then interconnecting them using block diagram algebra.

Without loss of generality, the dynamical structure function of a subsystem characterized by the zero-intricacy realization (A, B, C, D) is determined if we let $p_1 \leq p$ be the rank of C and assume without loss of generality that the outputs $y = [y_1' \;\; y_2']'$, $y_1 \in \mathbb{R}^p$ and $y_2 \in \mathbb{R}^{(l-p)}$, are ordered so the first p rows of C are linearly independent, i.e.

$$C = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \tag{2.17}$$

with $C_1 \in \mathbb{R}^{p \times n}$ being full row rank. The dynamical structure function of the system with respect to y_1 is then given by a pair of $(l \times p)$ and $(l \times m)$ real rational matrix functions, $(\hat{Q}(s), \hat{P}(s))$, defined over the Laplace variable, $s \in \mathbb{C}$.

In order to determine the a view of the system from the perspective of the manifest variables, first create the $(n \times n)$ state transformation:

$$T = \begin{bmatrix} C_1' & E_1 \end{bmatrix}', \tag{2.18}$$

where $E_1 \in \mathbb{R}^{n \times (n-p)}$ is any basis of the null space of C_1, with

$$T^{-1} = \begin{bmatrix} R_1 & E_1 \end{bmatrix}, \tag{2.19}$$

where $R_1 = C_1'(C_1C_1')^{-1}$.

Change basis such that $z = Tx$, yielding $\hat{A} = TAT^{-1}$, $\hat{B} = TB$, $\hat{C} = CT^{-1}$, and $\hat{D} = D$, and partitioned commensurate with the block partitioning of T and T^{-1} to give

$$\begin{aligned}
\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} &= \begin{bmatrix} \hat{A}_{11} & \hat{A}_{12} \\ \hat{A}_{21} & \hat{A}_{22} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} \hat{B}_1 \\ \hat{B}_2 \end{bmatrix} u \\
\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} &= \begin{bmatrix} I & 0 \\ \hat{C}_{21} & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} \hat{D}_1 \\ \hat{D}_2 \end{bmatrix} u
\end{aligned} \tag{2.20}$$

Note that while it is easily seen that $C_1R_1 = I$ and $C_1E_1 = 0$, but the fact that $C_2E_1 = 0$ may demand some reflection. The reason this is true is because every row of C_2 is in the row space of C_1. If it were not so, then either the rank of C would be greater than p or C_1 would not be composed of p linearly independent rows. Being in the row space of C_1, each row in C_2 is thus also orthogonal to every vector in E_1, which spans the orthogonal complement of the row space of C_1.

Assume zero initial conditions, take Laplace transforms, and solve for Z_2, yielding

$$sZ_1 = \left[\hat{A}_{11} + \hat{A}_{12}(sI - \hat{A}_{22})^{-1}\hat{A}_{21}\right] Z_1 + \left[\hat{B}_1 + \hat{A}_{12}(sI - \hat{A}_{22})^{-1}\hat{B}_2\right] U \tag{2.21}$$

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} I & 0 \\ \hat{C}_{21} & 0 \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} + \begin{bmatrix} \hat{D}_1 \\ \hat{D}_2 \end{bmatrix} U \tag{2.22}$$

where Z, U, and Y denote the Laplace transforms of z, u, and y respectively.

For notational simplicity, define:

$$W(s) = \hat{A}_{11} + \hat{A}_{12}(sI - \hat{A}_{22})^{-1}\hat{A}_{21} \tag{2.23}$$

$$V(s) = \hat{B}_1 + \hat{A}_{12}(sI - \hat{A}_{22})^{-1}\hat{B}_2 \tag{2.24}$$

and let $D_W(s) = \text{diag}(W(s))$ be a diagonal matrix function composed of the diagonal entries of $W(s)$.

Define $Q(s) = (sI - D_W)^{-1}(W - D_W)$ and $P(s) = (sI - D_W)^{-1}V$ yielding

$$Z_1 = Q(s)Z_1 + P(s)U \tag{2.25}$$

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} I & 0 \\ \hat{C}_{21} & 0 \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} + \begin{bmatrix} \hat{D}_1 \\ \hat{D}_2 \end{bmatrix} U \tag{2.26}$$

Noting from (2.25) that $Z_1 = Y_1 - D_1U$, the dynamical structure function of a zero-intricacy state space realization of the form (2.1) with respect to y_1 is then given by:

$$\hat{Q}(s) = \begin{bmatrix} Q(s) \\ C_{21} \end{bmatrix}, \; \hat{P}(s) = \begin{bmatrix} P(s) + (I - Q(s))D_1 \\ D_2 - C_{21}D_1 \end{bmatrix} \tag{2.27}$$

which satisfies

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \hat{Q}(s)Y_1 + \hat{P}(s)U \tag{2.28}$$

Definition 4. *The* signal structure *of a system is denoted $\mathscr{W}$, with a vertex set $V(\mathscr{W})$ and edge set $E(\mathscr{W})$, [70]. The elements of a system's signal structure is defined to be:*

- $V(\mathscr{W}) = \{u_1, ..., u_m, y_{11}, ..., y_{1p_1}, y_{21}, ..., y_{2p_2}\}$, *each representing a manifest variable of the system with $p_2 = p - p_1$, and*
- $E(\mathscr{W})$ *contains an edge from $v_i \in V(\mathscr{W})$ to $v_j \in V(\mathscr{W})$ if the associated entry of $\overline{Q}$ and $\overline{P}$ is nonzero.*

Unlike the subsystem structure, the signal structure uses circular nodes to denote signals rather than systems, while the edges between these signals represent systems since it is a condensation graph of the signal flow representation of the complete computational structure.

A key aspect of the derivation of the dynamical structure function is the transformation of the form (2.18) used to put C in the desired form. This transformation has several important properties, detailed below.

The first property of the transformed system is that it is invariant to a change of basis on the hidden states, much in the way a transfer function is invariant to state transformations. This essentially means that transforming the internal structure of hidden states does not change the input-output dynamics of the hidden states. This concept is formalized in Lemma 1.

Lemma 1. (Invariance to a Class of Block Diagonal Transformations) *Given a system (A, B, C, D) of the form (2.20) with dynamical structure function $(\hat{Q}, \hat{P})$, then $(\hat{Q}, \hat{P})$ is invariant to block diagonal state transformations; that is, the set of systems characterized by block diagonal state transformations,*

$$\mathcal{S} = \{(MAM^{-1}, MB, CM^{-1}, D) \mid M = \begin{bmatrix} I_{p\times p} & 0 \\ 0 & M_{22} \end{bmatrix}\}$$

with M_{22} any invertible matrix of appropriate size, all share the same dynamical structure function, $(\hat{Q}, \hat{P})$.

Proof. Transforming the given system, $\overline{z} = Mz$, yields

$$\begin{aligned}
\begin{bmatrix} \dot{\overline{z}}_1 \\ \dot{\overline{z}}_2 \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12}M_{22}^{-1} \\ M_{22}A_{21} & M_{22}A_{22}M_{22}^{-1} \end{bmatrix} \begin{bmatrix} \overline{z}_1 \\ \overline{z}_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ M_{22}B_2 \end{bmatrix} u \\
\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} &= \begin{bmatrix} I & 0 \\ C_{21} & 0 \end{bmatrix} \begin{bmatrix} \overline{z}_1 \\ \overline{z}_2 \end{bmatrix} + \begin{bmatrix} D_1 \\ D_2 \end{bmatrix} u
\end{aligned} \tag{2.29}$$

which leads to

$$\begin{aligned}
W(s) &= A_{11} + A_{12}M_{22}^{-1}(sI - M_{22}A_{22}M_{22}^{-1})^{-1}M_{22}A_{21} \\
&= A_{11} + A_{12}(sI - A_{22})^{-1}A_{21},
\end{aligned} \tag{2.30}$$

$$\begin{aligned}
V(s) &= B_1 + A_{12}M_{22}^{-1}(sI - M_{22}A_{22}M_{22}^{-1})^{-1}M_{22}B_2 \\
&= B_1 + A_{12}(sI - A_{22})^{-1}B_2.
\end{aligned} \tag{2.31}$$

Since $W(s)$ and $V(s)$ are invariant to M_{22}, $(\hat{Q}, \hat{P})$ also remain unchanged with respect to M_{22}. □

Note that Lemma 1 shows that the dynamical structure function is invariant to transformations on the manner in which hidden states interact with other hidden states on not how hidden states interact with measured states.

The second property of the state transformation (2.18) is that the choice of basis of the null space does not change the dynamical structure function. This is important because it means an optimal basis for the determination of the dynamical structure function is not required and, more importantly, that the dynamical structure function only depends on the states chosen to represent the view of the system. Note that this means the dynamical structure function is sensitive to the choice of C_1 in (2.17). This concept is formalized in Theorem 2.

Theorem 2. (Invariance to Basis of the Null Space) *Given a system* (A, B, C, D) *as in (2.1), consider two distinct bases of the null space of* C, $E \neq \overline{E}$, *with corresponding state transformations:*

$$T = \begin{bmatrix} C_1 \\ E' \end{bmatrix}, \quad \overline{T} = \begin{bmatrix} C_1 \\ \overline{E}' \end{bmatrix},$$

as in (2.18), and each leading to its corresponding dynamical structure function, $(\hat{Q}, \hat{P})$ *and* $(\overline{Q}, \overline{P})$ *as in (2.27). Then* $(\hat{Q}, \hat{P}) = (\overline{Q}, \overline{P})$.

Proof. Let $z = Tx$ and $\overline{z} = \overline{T}x$. Then $\overline{z} = \overline{T}T^{-1}z$:

$$\overline{T}T^{-1} = \begin{bmatrix} C_1 \\ \overline{E}' \end{bmatrix} \begin{bmatrix} R_1 & E \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & \overline{E}'E \end{bmatrix} \tag{2.32}$$

where $R_1 = C_1'(C_1C_1')^{-1}$. The block diagonal structure of $\overline{T}T^{-1}$ then ensures, by Lemma 1, that the dynamical structure function produced for $\overline{z}$ is the same as that for z, i.e. $(\hat{Q}, \hat{P}) = (\overline{Q}, \overline{P})$. □

The last property explored here is the invariance of the dynamical structure function to state permutations, this means that permuting the states of the dynamical structure function reveals the same dynamical structure function with a renumbering of the measured states. This concept is formalized in Theorem 3.

Theorem 3. (Invariance to State Permutations) *Consider a system as in (2.1) with state matrices* (A, B, C, D) *and dynamical structure function* $(\hat{Q}, \hat{P})$. *Then* $(\hat{Q}, \hat{P})$ *is invariant to state permutations; that is, the set of systems characterized by state permutations,*

$$\mathcal{S} = \{(MAM^{-1}, MB, CM^{-1}, D) \mid M \text{ is a permutation matrix}\},$$

all share the same dynamical structure function, $(\hat{Q}, \hat{P})$, *up to a permutation.*

Proof. The result follows from the fact that that the state transformation selected in Step 1 of constructing the dynamical structure function transforms each system in the set $\mathcal{S}$ to the same system for Step 2; the resulting dynamical structure function is thus the same, up to a permutation. To see this, consider the transformation T constructed for the unpermuted system, (A, B, C, D):

$$T = \begin{bmatrix} C_1 \\ E_1' \end{bmatrix}, \quad T^{-1} = \begin{bmatrix} R_1 & E_1 \end{bmatrix},$$

and compare with the transformation T_M constructed for any permuted system, $(MAM^{-1}, MB, CM^{-1}, D)$:

$$T_M = \begin{bmatrix} C_1 \\ E_1' \end{bmatrix} M^{-1} = TM^{-1},$$
$$T_M^{-1} = M \begin{bmatrix} R_1 & E_1 \end{bmatrix} = MT^{-1}.$$

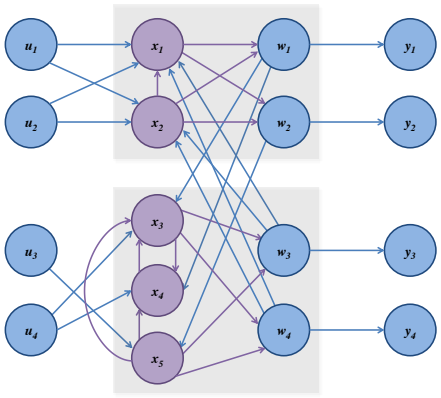
Applying each set of transformations to their respective systems yields the same transformed system, thus producing the same dynamical structure function:

$$(T_M MAM^{-1}T_M^{-1}, T_M MB, CM^{-1}T_M^{-1}, D) = (TAT^{-1}, TB, CT^{-1}, D).$$

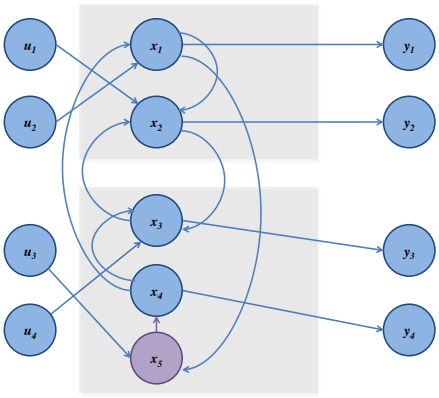
□

Example 5. *Given the generalized state space model in (2.6) with complete computational structure shown in Figure 2.1, the dynamical structure function is given in (2.33), the procedure for determining the corresponding signal structure from a system's generalized state space model is then outlined in Figure 2.6.*

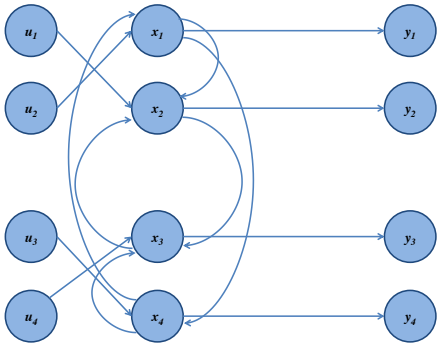
$$Q = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{s+1} \\ \frac{1}{s+2} & 0 & \frac{1}{s+2} & 0 \\ 0 & \frac{2}{s+1} & 0 & \frac{1}{s+1} \\ 0 & 0 & \frac{1}{(s+2)(s+3)} & 0 \end{bmatrix}, \; P = \begin{bmatrix} 0 & \frac{1}{s+1} & 0 & 0 \\ \frac{1}{s+2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{s+1} \\ 0 & 0 & \frac{1}{(s+2)(s+3)} & 0 \end{bmatrix} \tag{2.33}$$



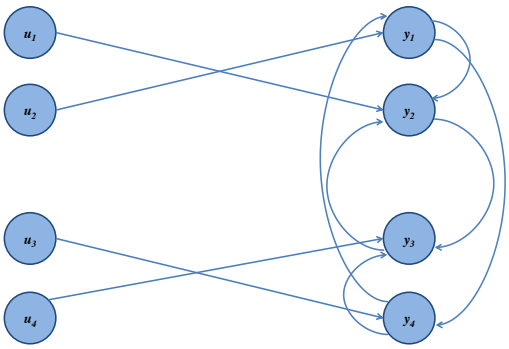
(a) Step 1: Begin with a generalized Complete Computational Structure.



(b) Step 2: Transform each subsystem so that $C_1 = \begin{bmatrix} I & 0 \end{bmatrix}$



(c) Step 3: Remove hidden nodes from the subsystems, maintaining pathways from manifest variable i to manifest variable j through the removed hidden nodes.



(d) Step 4: Rename manifest state variables, x, to their corresponding output variables, y, while removing the edges from manifest states to outputs along with the corresponding node.

Figure 2.6: Signal structure, built from a system's complete computational structure

2.5 Relationship Among System Representations

The transfer function associated with a given dynamical structure function is given by

$$H(s) = (I - Q)^{-1}P \tag{2.34}$$

which is found easily from (2.28) when Y_2 is of length zero. We note that $I - Q$ is invertible since Q is a square, hollow transfer function matrix, so $I - Q$ will always have full rank.

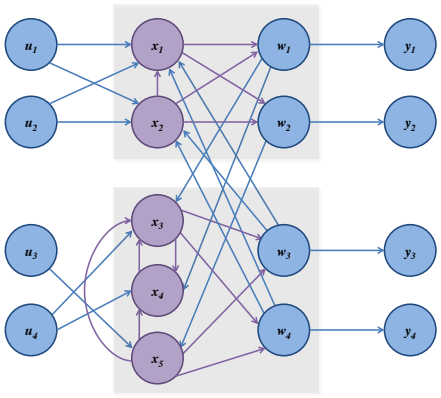
Necessary and sufficient conditions for determing a dynamical structure function given a system's transfer function were developed in [7].

Comparing the signal structure to subsystem structure, Example 6 shows that it is possible for a system's signal structure to be consistent with two or more subsystem structure representations. Example 7 shows it is also possible for a system's subsystem structure to be consistent with two or more dynamical structure functions. The implication of this result is that these two partial structure system representations denote two different notions of structure within a system.

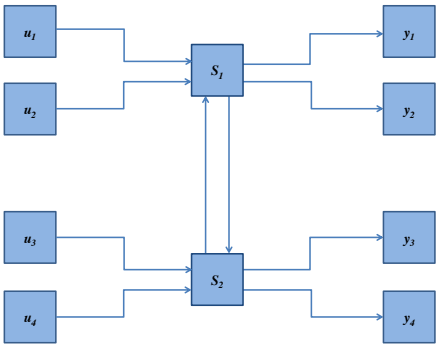
Example 6. *Given the complete computational structure shown in Figure 2.7a, the associated subsystem structure was found to be that shown in Figure 2.7b. Given a second complete computational structure in Figure 2.7c, the associated subsystem structure is shown in Figure 2.7d. Note that this complete computational structure is the same structure as the computational structure of the zero-intricacy realization of Figure 2.7a.*

The two structures are distinguished by the fact that the zero-intricacy structure in Figure 2.2 is a computational structure, meaning that it is not complete and the associated complete computational structure required auxiliary variables to model various compositions of functions. The complete computational structure given in Figure 2.7c, however, has the same structure, but no auxiliary variables were utilized for composition of functions, so the structure is considered complete.

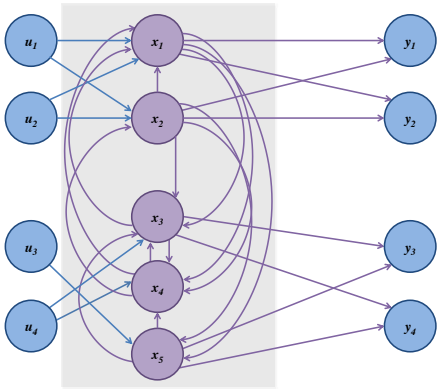
The corresponding signal structure for both complete computational structures is then given in Figure 2.7e, thus we have shown that it is possible for a single signal structure to be consistent with multiple subsystem structures.



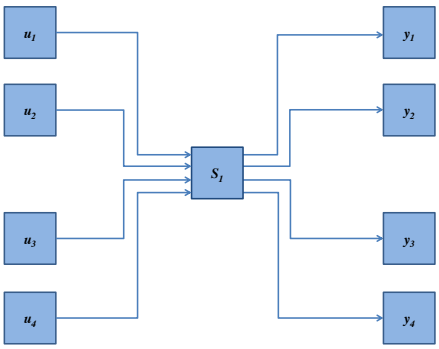
(a) The complete computational structure of a system.



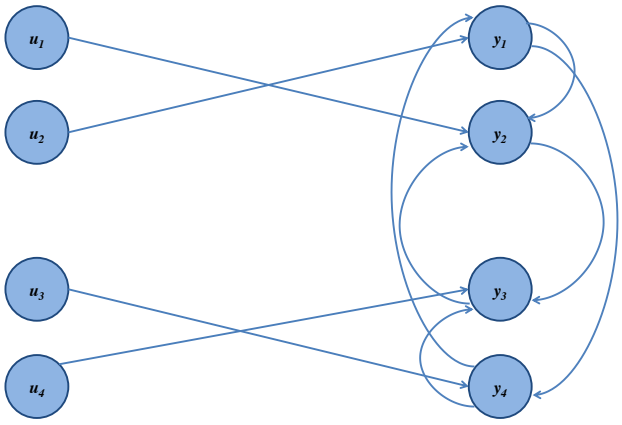
(b) This yields a subsystem structure of two systems in feedback.



(c) A complete computational structure without any intricacy variables.



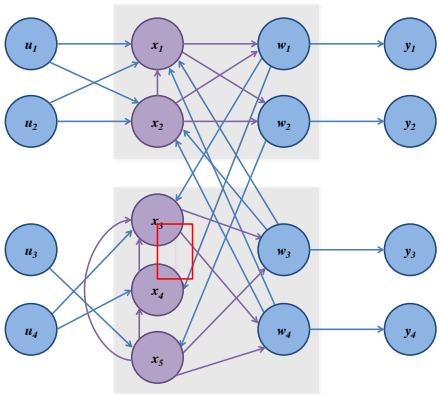
(d) This yields a new subsystem structure with a single subsystem.



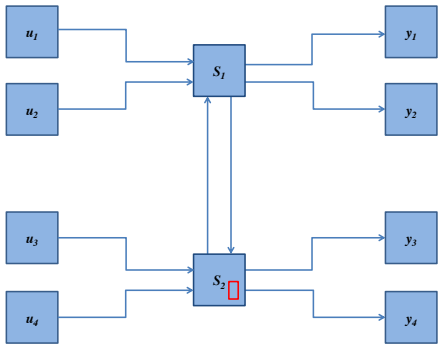
(e) The signal structure is the same for both complete computational structures.

Figure 2.7: Signal structure, Figure 2.7e, consistent with two subsystem structures, Figures 2.7b and 2.7d

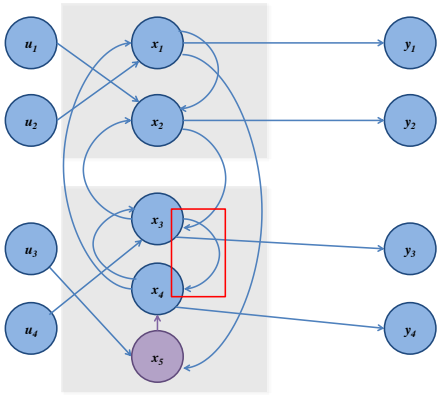
Example 7. *Given the complete computational structure in Figure 2.8a, which is the complete computational structure from Figure 2.1 with an edge removed (highlighted in red), the associated subsystem structure (shown in Figure 2.8b) does not change. Transforming the system to get* $C = \begin{bmatrix} I & 0 \end{bmatrix}$ *in each subsystem, yields the structure given in Figure 2.8c, which is similar to the transformed structure given in Figure 2.6b, with an extra edge, highlighted in red. The associated signal structure is then given in Figure 2.8d, also containing an extra edge meaning the subsystem structure given in Figure 2.8b is consistent with multiple signal structures.*



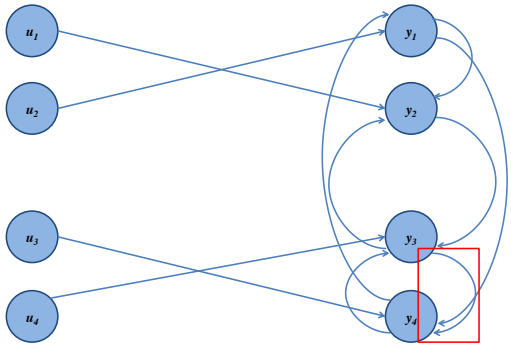
(a) Remove an edge from the complete computational structure.



(b) The subsystem structure remains the same since, with two subsystems in feedback.



(c) The transformed internal structure with an extra edge.



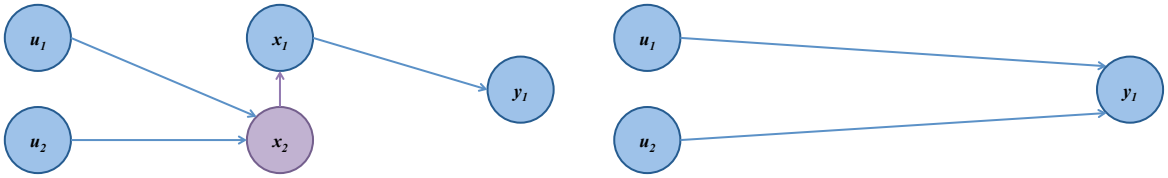
(d) The signal structure with an extra edge that does not exist in Figure 2.7e.

Figure 2.8: Subsystem structure, Figure 2.8b, consistent with two signal structures, Figures 2.7e and 2.8d

One of the properties of the signal structure of a system that distinguishes it from the subsystem structure is known as shared hidden states.

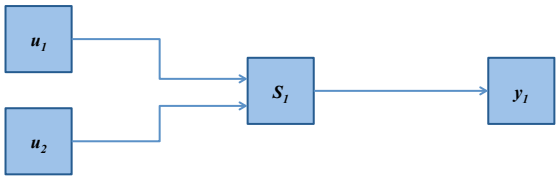
Definition 5. *A shared hidden state is a state within a system that is not manifest, i.e. that is part of the hidden structure, that has either multiple pathways from it that lead towards a manifest structure or multiple pathways that come from manifest structure or both.*

When a system contains a shared hidden state, the associated signal structure is agnostic to that state, meaning that it allows for hidden states to be shared across system edges. In contrast, the subsystem structure does not allow for hidden states to be shared across systems. Therefore, when shared hidden states exist in a system the signal structure contains more structural information than the subsystem structure as shown in Figure 2.9. Moreover, since the signal structure is agnostic to shared hidden states, the process of determining a unique dynamical structure function from a system's transfer function has reasonable conditions unlike the subsystem structure of the system.



(a) Complete computational structure with shared hidden node.

(b) Signal structure containing paths that represent two separate systems.



(c) Subsystem structure containing only a single subsystem.

Figure 2.9: Shared Hidden State

The dynamical structure function of a system is uniquely defined given a zero-intricacy state space model, as derived previously in this section. Determining a unique state space

model given a system's dynamical structure function is an ill-posed problem, though a procedure for determining a minimal state space realization given a dynamical structure function (Q, P) was given in [72].

Chapter 3

Identifiability Conditions of the Dynamical Structure Function

(Published at CDC 2012 as "Dynamical Structure Function Identifiability Conditions Enabling Signal Structure Reconstruction")

Abstract

Networks of controlled dynamical systems exhibit a variety of interconnection patterns that could be interpreted as the structure of the system. One such interpretation of system structure is a system's signal structure, characterized as the open-loop causal dependencies among manifest variables and represented by its dynamical structure function. Although this notion of structure is among the weakest available, previous work has shown that if no a priori structural information is known about the system, not even the Boolean structure of the dynamical structure function is identifiable. Consequently, one method previously suggested for obtaining the necessary a priori structural information is to leverage knowledge about target specificity of the controlled inputs. This work extends these results to demonstrate precisely the a priori structural information that is both necessary and sufficient to reconstruct the network from input-output data. This extension is important because it significantly broadens the applicability of the identifiability conditions, enabling the design of network reconstruction experiments that were previously impossible due to practical constraints on the types of actuation mechanisms available to the engineer or scientist. The work is motivated by the proteomics problem of reconstructing the Per-Arnt-Sim Kinase pathway used in the metabolism of sugars.

3.1 Introduction: Network Reconstruction

Two fundamental properties characterizing networks of controlled dynamical systems include their overall, dynamic behavior and their network structure. Since interconnections of systems are themselves systems, standard mathematical representations of systems can be used to describe networks of controlled dynamical systems. Nevertheless, while different representations may describe the same overall dynamic behavior of the network, they can convey very different information about the network's underlying structure.

Consider, for example, an n^{th} order, causal linear time-invariant system with m inputs and p outputs. This system may be described both by a $p \times m$ transfer function (TF) matrix $G(s)$ and a state space realization characterized by the matrices (A, B, C, D). If $G(s) = C(sI - A)^{-1}B + D$, then both of these representations describe the overall dynamic behavior of the system. Nevertheless, these representations carry very different information about the system's internal network structure.

In particular, the sparsity pattern of G reveals closed-loop dependencies of the outputs on inputs. Thus, for example, if G were diagonal, then any dependencies that output i may have on any other input, other than input i, must be exactly cancelled internally within the system. This sparsity structure of the TF is the weakest notion of system structure that we typically consider.

On the other hand, the state space realization describes detailed information about the dependencies among inputs, state variables, and outputs. This offers a much stronger understanding of system structure; it conveys everything revealed by the sparsity structure of the TF and much more about system structure. For example, knowing the state realization can determine whether a system with a diagonal TF G is truly decoupled, or whether off-diagonal dependencies between inputs and outputs are externally hidden by exact cancellations within the system.

Because different representations of the same system describe different amounts of structural information, they need different experimental conditions to be correctly identified.

For example, a rich theory of system identification has detailed sufficiency of excitation and sample complexity requirements necessary to determine a system's TF from input-output data [71]. Discovering richer structural information than the sparsity structure of the TF, however, demands additional a priori knowledge about the system. For example, if one not only met the requirements to identify a system's TF from input-output data, but also knew that the measured outputs were, in fact, the system's state variables, then the state space representation of the system could also be identified.

For many applications, the experimental burden necessary to identify a network's state space realization is prohibitive, while the burden necessary to identify the associated TF may be reasonable. In these situations, and when additional structural information about the system is desired, another representation of the system may be employed that describes more structural information than the TF yet less than the state space realization (and thus incurring less of an experimental burden). The dynamical structure function (DSF) of a linear time-invariant system is just such a *partial-structure* representation.

This paper describes the experimental burden necessary and sufficient to identify a system's DSF. This burden is characterized in terms of parts of the DSF that must be known a priori so that knowledge of the TF, identified from input-output data, would then uniquely specify the remaining parts of the DSF. Previous work has shown that if only the input-output data necessary to identify the system's TF are known a priori, then not even the Boolean structure, which denotes the presence of a causal relationships without its effects, of the DSF can be reconstructed. On the other hand, if one additionally knows that each input, u_i, exhibits *target specificity*, in that it only affects outputs y_j though its associated output, y_i, then the DSF can be uniquely reconstructed from sufficiently informative input-output data or knowledge of the system TF [24].

In this paper we show that, while target specificity is a sufficient condition for reconstruction of the DSF, it is not necessary. By providing a complete set of necessary and sufficient conditions for reconstruction of a system's DSF, we significantly broaden the

applicability of the associated reconstruction results [24, 60, 70, 73] . The next section details mathematical preliminaries concerning DSF as a partial-structure representation of a system. The main result then follows, with necessary and sufficient identifiability conditions for reconstructing a system's DSF. The last section motivates the utility of these new results with a network reconstruction problem from proteomics where target specificity can not be guaranteed. Conclusions follow.

3.2 Dynamical Structure Functions

This section gives an overview of the derivation of DSF as discussed in [24]. To see how the DSF is derived, consider the system given by:

$$\begin{aligned} \begin{bmatrix} \dot{y} \\ \dot{x} \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \\ y &= \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix}. \end{aligned} \tag{3.1}$$

Note that $C = \begin{bmatrix} I & 0 \end{bmatrix}$ allows the variables to be separated into the measured states, y, and the unmeasured states, x.

Equation (3.1) describes the state space realization of the system, which contains information about the dependency among input, state, and output variables and defines both the structure and dynamics of the entire network.

The next step in the derivation of the DSF is to take Laplace Transforms of the signals in (3.1). Assuming zero initial conditions we get:

$$\begin{bmatrix} sY \\ sX \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} Y \\ X \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} U \tag{3.2}$$

Solving for X, gives:

$$X = (sI - A_{22})^{-1} A_{21} Y + (sI - A_{22})^{-1} B_2 U$$

Substituting into the first equation of (3.2) then yields

$$sY = WY + VU$$

where $W = A_{11} + A_{12} (sI - A_{22})^{-1} A_{21}$ and $V = A_{12} (sI - A_{22})^{-1} B_2 + B_1$. Let D be a matrix with the diagonal term of W, i.e. $D = \text{diag}(W_{11}, W_{22}, ..., W_{pp})$. Then,

$$(sI - D) Y = (W - D) Y + VU$$

Note that $sI - D$ is always invertible since D is always proper. We then have:

$$Y = QY + PU \tag{3.3}$$

where

$$Q = (sI - D)^{-1} (W - D) \tag{3.4}$$

and

$$P = (sI - D)^{-1} V \tag{3.5}$$

Note that since $W - D$ is a hollow matrix (a matrix with zeros along the diagonal), then Q is also a hollow matrix.

The matrix Q is a matrix of strictly proper TF from Y_i to Y_j, $i \neq j$ relating each measured signal to all other measured signals. Likewise, P is a matrix of strictly proper TF from each input to each output without depending on any additional measured state Y_i.

Together, the pair $(Q(s), P(s))$ is the DSF for the system, a partial structure representation of the system.

Definition 6. *Given the system (3.1), we define the* dynamical structure function *of the system to be (Q, P), where Q and P are the* internal structure *and* control structure*, respectively, and are given as in (3.4) and (3.5).*

3.2.1 Meaning of the Dynamical Structure Function

The DSF describes the network structure of the system (3.1) in the sense that the matrix Q can be interpreted as the weighted adjacency matrix of a directed graph, an example of which is shown in Figure 3.1, indicating the causal relationships between measured states. Also, P is the weighted adjacency matrix of a directed graph indicating the causal relationships between inputs and measured states. The weights on the edges of this graph are TF between relevant variables. This graphical representation of the system is referred to as the signal structure of the system.

Note that the TF matrix of the graph in Figure 3.1 would be full because every input affects each output, whether it be directly or indirectly. The DSF, on the other hand, contains more information about the structure of the system since each edge represents the open-loop causal dependencies of the manifest variables.

The way in which a DSF represents the open-loop causal dependencies of a system is seen via the following theorem:

Theorem 4. *Let (A, B) be the matrices from the state-space representation of an LTI system, and (Q, P) its dynamical structure function.*

1. *If Q_{ij} is nonzero, then either A_{ij} is nonzero or there exists a sequence $k_1, k_2, ...$ of indices corresponding to hidden states such that $A(i, k_1), A(k_1, k_2), ..., A(k_{m-1}, k_m), A(k_m, j)$ are nonzero*

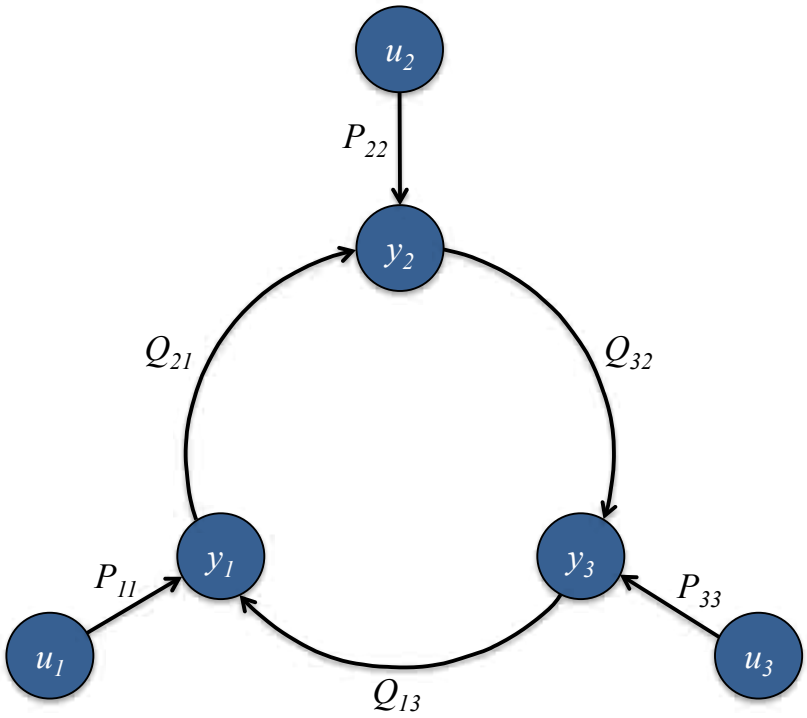


Figure 3.1: Signal Structure representation of a system with three inputs and three outputs. Note that the transfer function for this system is fully connected, while the dynamical structure function, and its associated signal structure, exhibits a particular ring structure.

2. *If P_{ij} is nonzero, then either B_{ij} is nonzero or there exists a sequence $k_1, k_2, ...$ of indices corresponding to hidden states such that $A(i,k_1), A(k_1,k_2), ..., A(k_{m-1},k_m), A(k_m,j)$ and $B(k_m,j)$ are nonzero*

Proof. We proceed with a proof by contradiction. Assume that Q_{ij} is nonzero, but that A_{ij} is zero and there does not exist a sequence $k_1, k_2, ...$ of indices corresponding to hidden states such that $A(i,k_1), A(k_1,k_2), ..., A(k_{m-1},k_m), A(k_m,j)$ are nonzero. Then, we see that:

$$W_{ij} = A_{ij} + \left(A_{12}\left(sI - A_{22}\right)^{-1} A_{21}\right)_{ij} = 0$$

From the definition of Q in (3.4), we note that the off-diagonal values of Q are nonzero (zero) when the off-diagonal values of W are nonzero (zero). Thus,

$$W_{ij} = 0 \implies Q_{ij} = 0$$

which gives us a contradiction. The proof for part (b) is similar. □

3.2.2 Relationship of the Dynamical Structure Function to a System's Transfer Function

Given any system (3.1), its DSF, like its TF, is uniquely specified. However, although the system's TF contains the dynamics of the system, it yields no information about the structure of the network . The direct relationship between a system's DSF and it's TF can be defined as:

Lemma 2. *[24] The transfer function, G, of the system (3.1), is related to its dynamical structure, (Q, P), by*

$$G = (I - Q)^{-1}P \tag{3.6}$$

This follows from (3.3) and $Y = GU$.

3.2.3 Signal Structure Reconstruction

Network reconstruction considers the problem of finding the DSF of a given system, which is consistent with its TF. This makes it similar to the realization problem, which is concerned with finding the state-space description that is consistent with a system's TF, shown in Figure 3.2.

The problem of reconstruction is ill-posed, however, since there are many different state space realizations that are consistent with a given TF, G, or DSF, (Q, P). The only way to reconstruct the system and yield the state space model is if the system has full state feedback, meaning that every node in the system is measured. Since this is not generally possible, various assumptions need to be made about the system a priori or more information about the system is needed before the state space realization can be determined.

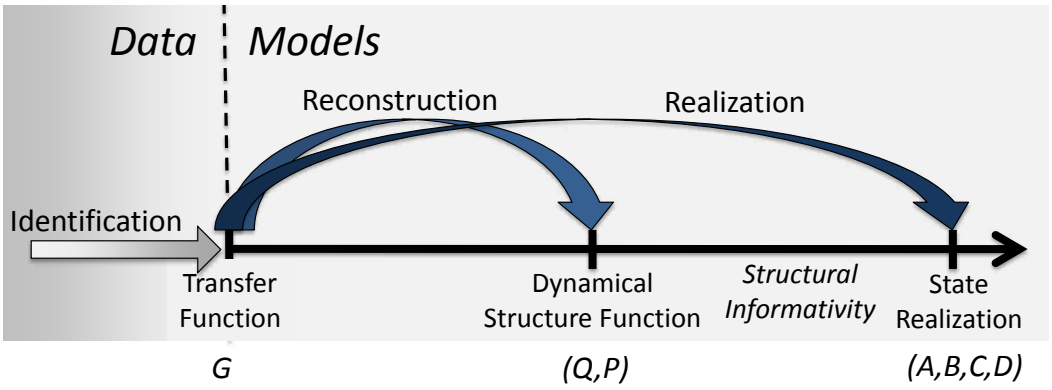


Figure 3.2: Different types of models of the same system describe different amounts of structural information. The reconstruction problem, like the realization problem, requires additional information about the system above and beyond the input-output data necessary to identify the transfer function.

The same is true when attempting to determine a system's DSF from a system's TF. In fact, defining A' to be the conjugate transpose of the matrix A, we get the following relationship between the DSF and TF of a system:

Lemma 3. *[24] Given a transfer function G, the set $\mathscr{S}_{\mathscr{G}}$ of all dynamical structure functions consistent with G can be parameterized by a $p \times p$ internal structure function $\tilde{Q}$ and is given by*

$$\mathscr{S}_{\mathscr{G}} = \left\{ (Q,P) : \begin{bmatrix} Q' \\ P' \end{bmatrix} = \begin{bmatrix} 0 \\ G' \end{bmatrix} + \begin{bmatrix} I \\ -G' \end{bmatrix} \tilde{Q}', \tilde{Q} \in \mathscr{Q} \right\}, \tag{3.7}$$

where $\mathscr{Q}$ is the set of internal structure functions. Moreover, the set $\mathscr{S}_{\mathscr{G}}$ has $p^2 - p$ degrees of freedom.

Thus, the reconstruction problem can be characterized as in [24] with the following theorem:

Theorem 5. *Given any $p \times m$ transfer function G, with $p > 1$ and no other information about the system, dynamical and boolean reconstruction is not possible. Moreover, for any internal structure Q, there is a dynamical structure function (Q,P) that is consistent with G.*

However, since the DSF (Q, P) is a partial structure representation of the network, less information is needed for the DSF to be reconstructed than for the entire state space to be realized.

The following corollary from previous work indicates what partial-structure information, which refers to knowledge of some of the elements of Q or P, is sufficient for dynamical structure reconstruction.

Corollary 1. *[24] If $m = p$, G is full rank, and there is no information about the internal structure, Q, then the dynamical structure can be reconstructed if each input controls a measured state independently, that is, without loss of generality, the inputs can be indexed such that P is diagonal.*

This section served to describe the reconstruction methodology and the basic requirements for reconstruction of the DSF of a system such as (3.1). Equipped with this description, we are now prepared to extend the details discussed in this section in order to allow for systems whose states cannot be independently controlled. These systems have a P matrix that is not diagonal, hence previous methods have to be extended to accommodate them.

3.3 Main Result

Identifiability conditions fundamentally concern the definition of a map from model parameters to data and ensuring that it is injective. In this way, a particular set of parameters is uniquely specified by the data, identifying the correct model from the set of models under consideration.

Identifying a system's DSF from data involves the standard issues related to identifying a TF from data (sufficiency of excitation, etc.), along with additional issues related to the fact that many DSF generate the same TF (consider Lemma 3). In the sequel we will ignore the standard issues and focus on the additional identifiability issues unique to DSF. Consequently we will assume that the system's TF has been successfully identified from data and focus on necessary and sufficient conditions for then recovering the DSF. To accomplish this, we

will construct the map from the elements of the DSF to the associated TF and establish conditions ensuring this map is injective.

To facilitate the discussion, we introduce the following notation. Let $A \in \mathbb{C}^{n\times m}$ and $B \in \mathbb{C}^{k\times l}$. Then:

- blckdiag(A, B) is the block diagonal matrix given by

$$\begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix},$$

- a_i is the i^{th} column of matrix A,
- A_{-i} is the matrix A without it's i^{th} column,
- a_{ij} is the $(i,\, j)^{th}$ entry of matrix A,
- A' is the conjugate transpose of matrix A,
- $\mathcal{R}(A)$ is the range of A,
- $\overrightarrow{a}$ is the vector stack of the columns of A, given by

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}$$

- and $\overleftarrow{a}$ is the vector stack of the columns of A'.

The construction of a map from elements of the DSF to the associated TF begins by rearranging the relationship from Lemma 2 in Equation (3.6) to yield:

$$\begin{bmatrix} I & G' \end{bmatrix} \begin{bmatrix} P' \\ Q' \end{bmatrix} = G' \tag{3.8}$$

Noting that

$$AX = B \iff \text{blckdiag}(\text{A}, ..., \text{A})\vec{x} = \vec{b}$$

and defining $X = \begin{bmatrix} P' & Q' \end{bmatrix}$ then allows us to rewrite Equation (3.8) as

$$\begin{bmatrix} I & \text{blckdiag}(G', ..., G') \end{bmatrix} \overleftarrow{x} = \overleftarrow{g}\,. \tag{3.9}$$

Further noting that since the diagonal elements of Q are identically zero and the dimensions of P, Q, and G are $p \times m$, $p \times p$, and $p \times m$ respectively, then exactly p elements of $\overleftarrow{x}$ are always zero. Abusing notation, we can then redefine $\overleftarrow{x}$ to remove these zero elements, reducing Equation (3.9) to the following:

$$\begin{bmatrix} I & \text{blckdiag}(G'_{-1}, G'_{-2}, ..., G'_{-p}) \end{bmatrix} \overleftarrow{x} = \overleftarrow{g}\,. \tag{3.10}$$

Equation (3.10) reveals the mapping from elements of the DSF, contained in $\overleftarrow{x}$, to its associated TF, represented by $\overleftarrow{g}$. The mapping is clearly a linear transformation represented by the matrix operator $\begin{bmatrix} I & \text{blckdiag}(G'_{-1}, G'_{-2}, ..., G'_{-p}) \end{bmatrix}$. This matrix has dimensions $(pm) \times (pm + p^2 - p)$, and thus the transformation is certainly not injective. This is why not even the Boolean structure of a system's DSF can be identified – even from perfect information about the system's TF – without additional a priori structural information.

Identifiability conditions will thus be established by determining which elements of $\overleftarrow{x}$ must be known a priori in order to reduce the corresponding transformation to an injective map. To accomplish this, consider the $(pm + p^2 - p) \times k$ transformation T such that

$$\overleftarrow{x} = Tz \tag{3.11}$$

where z is an arbitrary vector of size k. The following lemma describes technical conditions on T establishing necessary and sufficient identifiability conditions for DSF reconstruction.

Lemma 4. *Let*

$$M = LT, \tag{3.12}$$

where $L = \begin{bmatrix} I & \text{blckdiag}(G'_{-1}, G'_{-2}, ..., G'_{-p}) \end{bmatrix}$ *and* T *is a* $(pm + p^2 - p) \times k$ *matrix operator as in Equation (3.11). Then* M *is injective if and only if*

1. $k \leq pm$, *and*
2. $\text{rank}(T) = k$ *(i.e.* T *is injective).*

Proof. Since $\begin{bmatrix} I & \text{blckdiag}(G'_{-1}, G'_{-2}, ..., G'_{-p}) \end{bmatrix}$ has rank pm, $\text{rank}(M) = \min(pm, \text{rank}(T))$. If $\text{rank}(T) > pm$, implying $k > pm$, then M is clearly not injective. If $\text{rank}(T) \leq pm$, then $\text{rank}(M) = \text{rank}(T)$ and M will be injective if and only if $k = \text{rank}(T)$. □

Theorem 6. (Identifiability Conditions) *Given a system characterized by the transfer function* G*, its DSF* (Q, P) *can be identified if and only if*

1. M*, defined as in Equation (3.12), is injective, and*
2. $\overleftarrow{g} \in \mathcal{R}(M)$.

Proof. The proof follows immediately from the observation that M is the mapping from unidentified model parameters to the system TF. Under these conditions one can clearly solve for z given G and then construct the DSF from $\overleftarrow{x}$, where $\overleftarrow{x} = Tz$, and T is precisely the a priori system information that is necessary and sufficient for reconstruction. □

We will now illustrate this reconstruction result on some simple examples.

Example 8. *Consider a system with square TF given by*

$$G = \begin{bmatrix} G_{11} & G_{12} & ... & G_{1p} \\ G_{21} & G_{22} & & G_{2p} \\ \vdots & & \ddots & \vdots \\ G_{p1} & G_{p2} & ... & G_{pp} \end{bmatrix}.$$

Previous work has shown that if G is full rank and it is known, a priori, that the control structure P is diagonal that reconstruction is possible [24]. Here we validate that claim by demonstrating that the associated T matrix becomes:

$$\begin{bmatrix} P_{11} \\ P_{12} \\ \vdots \\ P_{21} \\ P_{22} \\ \vdots \\ P_{pp} \\ Q_{12} \\ \vdots \\ Qp(p-1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & ... & 0 & 0 \\ 0 & 0 & ... & 0 & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & ... & 0 & 0 \\ 0 & 1 & ... & 0 & 0 \\ \vdots & \ddots & & \ddots & \vdots \\ 0 & ... & 1 & ... & 0 \\ 0 & ... & 0 & 1 & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & ... & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{11} \\ P_{22} \\ \vdots \\ P_{pp} \\ Q_{12} \\ \vdots \\ Q_{p(p-1)} \end{bmatrix}$$

yielding the operator $M = LT$ as:

$$M = \begin{bmatrix} e_1 & 0 & 0 & G'_{-1} & ... & 0 \\ 0 & \ddots & 0 & 0 & \ddots & 0 \\ 0 & ... & e_p & 0 & ... & G'_{-p} \end{bmatrix}$$

where e_i is a zero vector with 1 in the i^{th} position. Note that M is a square matrix with dimensions $p^2 \times p^2$ and will be invertible provided G is full rank, thus enabling reconstruction.

Example 9. *Given the following TF of a system:*

$$G = \begin{bmatrix} \frac{s+2}{s^2+3s+1} & -\frac{s^2+3s+3}{(s+2)(s^2+3s+1)} \\ \frac{s+2}{(s+1)(s^2+3s+1)} & \frac{s^2+s-1}{(s+1)(s^2+3s+1)} \end{bmatrix}$$

We attempt to find the DSF (Q, P) of the system:

$$Q = \begin{bmatrix} 0 & Q_{12} \\ Q_{21} & 0 \end{bmatrix} \text{ and } P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$$

yielding the vector of unknowns $\vec{x} = \begin{bmatrix} P_{11} & P_{12} & P_{21} & P_{22} & Q_{12} & Q_{21} \end{bmatrix}'$. *This gives us the system of equations of the form* $L\vec{x} = \vec{b}$:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \frac{s+2}{(s+1)(s^2+3s+1)} & 0 \\ 0 & 1 & 0 & 0 & \frac{s^2+s-1}{(s+1)(s^2+3s+1)} & 0 \\ 0 & 0 & 1 & 0 & 0 & \frac{s+2}{s^2+3s+1} \\ 0 & 0 & 0 & 1 & 0 & -\frac{s^2+3s+3}{(s+2)(s^2+3s+1)} \end{bmatrix} \begin{bmatrix} P_{11} \\ P_{12} \\ P_{21} \\ P_{22} \\ Q_{12} \\ Q_{21} \end{bmatrix} = \begin{bmatrix} \frac{s+2}{s^2+3s+1} \\ -\frac{s^2+3s+3}{(s+2)(s^2+3s+1)} \\ \frac{s+2}{(s+1)(s^2+3s+1)} \\ \frac{s^2+s-1}{(s+1)(s^2+3s+1)} \end{bmatrix}$$

Without additional information a priori structural information, we can not reconstruct. Suppose, however, that we know a priori that P takes the form:

$$P = \begin{bmatrix} P_{11} & -P_{11} \\ 0 & P_{22} \end{bmatrix}$$

Note that this non-diagonal P fails to meet the previous conditions for reconstruction [23, 24]. Nevetheless, the vector of unknowns $\vec{x}$ can then be decomposed into the form $T\vec{z}$ as

follows:

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } \vec{z} = \begin{bmatrix} P_{11} & P_{22} & Q_{12} & Q_{21} \end{bmatrix}'$$

Replacing $\vec{x}$ with $T\vec{z}$ above yields the system of equations of the form $M\vec{z} = \vec{b}$, where $M = LT$:

$$\begin{bmatrix} 1 & 0 & \frac{s+2}{(s+1)(s^2+3s+1)} & 0 \\ -1 & 0 & \frac{s^2+s-1}{(s+1)(s^2+3s+1)} & 0 \\ 0 & 0 & 0 & \frac{s+2}{s^2+3s+1} \\ 0 & 1 & 0 & -\frac{s^2+3s+3}{(s+2)(s^2+3s+1)} \end{bmatrix} \begin{bmatrix} P_{11} \\ P_{22} \\ Q_{12} \\ Q_{21} \end{bmatrix} = \begin{bmatrix} \frac{s+2}{s^2+3s+1} \\ -\frac{s^2+3s+3}{(s+2)(s^2+3s+1)} \\ \frac{s+2}{(s+1)(s^2+3s+1)} \\ \frac{s^2+s-1}{(s+1)(s^2+3s+1)} \end{bmatrix}$$

In this case M is full rank, from theorem 6 we know that the system is reconstructible. By solving for $\vec{x} = (M)^{-1}\vec{b}$ we get the DSF to be:

$$Q = \begin{bmatrix} 0 & \frac{1}{s+2} \\ \frac{1}{s+1} & 0 \end{bmatrix} \text{ and } P = \begin{bmatrix} \frac{1}{s+1} & -\frac{1}{s+1} \\ 0 & \frac{1}{s+2} \end{bmatrix}$$

3.4 Motivating Example: The PAS Kinase Pathway

An example of such a network is the Per-Arnt-Sim (PAS) Kinase Pathway. Human mutations in the PAS Kinase Pathway have recently been linked to the early development of type 2 diabetes [55]. The PAS Kinase Pathway is composed of proteins that interact in specific ways to direct the metabolism of sugars in eukaryotic cells. Each of these proteins have both an activated and a deactivated form that serves a distinct function within the network. The

identification of network structure in this system is an ideal application of signal structure theory.

Several PAS Kinase networks have been proposed such as in [32], so analysis of such a pathway with the DSF method would help to indicate flaws or validate proposed biological pathways. Yeast serves as a model biological organism for understanding the basic processes of life due to the ease of study and the conservation of many pathways. In fact, the best characterized PAS Kinase pathway, the Ugp1 pathway, was first identified in yeast [49].

One of the proposed networks for the PAS Kinase Ugp1 pathway is indicated in Fig. 3.3. In this pathway there are three proteins that are directly formed from a gene: PSK, Ugp1, and Glc7; the others are activated forms or complexes involving these three proteins. The species of interest in the pathway are Ugp1, Ugp1* and the Ugp1*Glc7. The asterisk implies an activated form of the protein, e.g Upg1* is the activated form of Ugp1 that is produced when PAS Kinase modifies the Ugp1 protein [49]. Once Ugp1 is activated, it partitions the use of cellular glucose towards structural components at the expense of storage carbohydrates [56]. The last species, Ugp1*Glc7, is theoretical and may be formed by a direct interaction between Ugp1* and Glc7 [49]. It is hypothesized that Ugp1* is deactivated by this process, however this needs to be verified. Other key network players include the Snf1 protein, which is required for the activation of PAS Kinase in response to available nutrients [26].

As shown, the current theoretical network involves ten species, with the majority of the pathway verified. It is not easy to directly perturb each of the three nodes of interest since PSK affects two of the observed nodes however, it is possible to create experiments that directly affect two of the species. These experiments consist of turning on or off a specified gene by modifying the yeast cell or its environment. This is commonly done through the use of a plasmid, a small circular piece of DNA inserted into the yeast cell that expresses the protein in response to external stimulus (such as the addition of a particular chemical to the growth media). The experiments for the PAS Kinase network include manipulation of the genes Glc7, Ugp1, and PSK. The plasmids with PSK will directly affect two observed

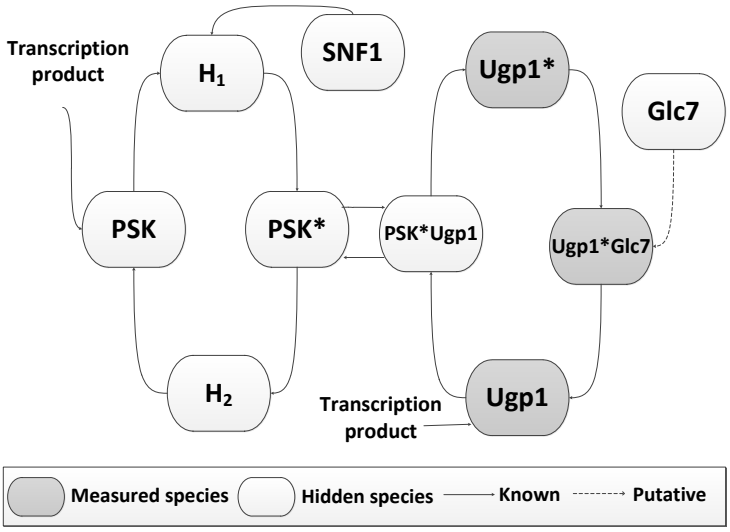


Figure 3.3: PAS Kinase Pathway with H_1 and H_2 representing networks of unobserved nodes

nodes: Ugp1* and Ugp1. However, this will be done in an equal amount; it will increase the activated form of Ugp1* while decreasing the inactive form, Ugp1. The experimental setup is shown below in Fig. 3.4.

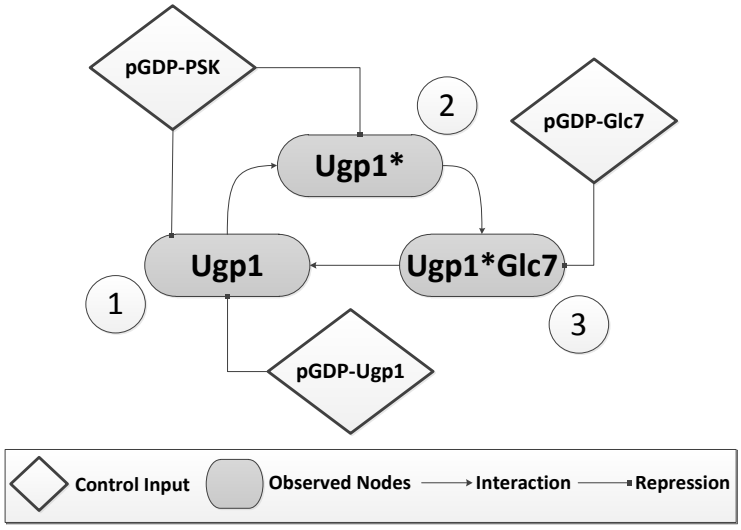


Figure 3.4: Experimental setup for PAS Kinase Pathway

As expected, the exact mechanisms by which phosphorylation and dephosphorylation occur are hidden in this formulation. As indicated earlier, previous formulation required

a direct perturbation for each observed node in any given network. However, methods or experimental conditions that independently perturb observed nodes in biological networks are usually not feasible. This becomes even more difficult to do when biological networks have several observed nodes; in many cases it is even impossible to independently perturb all the observed nodes. However, with the extensions indicated in Section 3.3, reconstruction is still possible despite multiple perturbations of observed nodes in a given experiment. This is demonstrated for the Pas Kinase pathway as indicated in Fig. 3.4

3.4.1 Reconstruction for PAS Kinase Pathway

We can define Q, P, and G for the pathway as follows:

$$G_{PAS} = \begin{bmatrix} G_1 & G_2 & G_3 \\ G_4 & G_5 & G_6 \\ G_7 & G_8 & G_9 \end{bmatrix}, \quad Q_{PAS} = \begin{bmatrix} 0 & Q_1 & Q_2 \\ Q_3 & 0 & Q_4 \\ Q_5 & Q_6 & 0 \end{bmatrix}, \quad P_{PAS} = \begin{bmatrix} P_1 & P_2 & P_3 \\ P_4 & P_5 & P_6 \\ P_7 & P_8 & P_9 \end{bmatrix} \tag{3.13}$$

However, from the experimental design indicated in Fig. 3.4, we know the true representation of the control matrix P is as follows:

$$P_{PAS} = \begin{bmatrix} -P_{PSK} & P_{Ugp1} & 0 \\ P_{PSK} & 0 & 0 \\ 0 & 0 & P_{Glc7} \end{bmatrix}$$

Where $P_{PSK} = P_1, P_{Ugp1} = P_2$, and $P_{Glc7} = P_9$. This true representation of P serves as the prior knowledge for this system. Given $L\vec{x} = \vec{b}$ such that:

$$\left[\begin{array}{c|cccccc} \mathrm{I} & G_4 & G_7 & 0 & 0 & 0 & 0 \\ & G_5 & G_8 & 0 & 0 & 0 & 0 \\ & G_6 & G_9 & 0 & 0 & 0 & 0 \\ & 0 & 0 & G_1 & G_7 & 0 & 0 \\ & 0 & 0 & G_2 & G_8 & 0 & 0 \\ & 0 & 0 & G_3 & G_9 & 0 & 0 \\ & 0 & 0 & 0 & 0 & G_1 & G_4 \\ & 0 & 0 & 0 & 0 & G_2 & G_5 \\ & 0 & 0 & 0 & 0 & G_3 & G_6 \end{array}\right] \times \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_9 \\ Q_1 \\ \vdots \\ Q_6 \end{bmatrix} = \begin{bmatrix} G_1 \\ G_2 \\ \vdots \\ G_9 \end{bmatrix}$$

This system has 15 unknowns and 9 equations, so it is easy to see that no unique solution exists as is. However, taking into account a priori information given the true structure of P, we can decompose $\vec{x}$ into $T\vec{z}$ as follows:

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and } \vec{z} = \begin{bmatrix} P_1 \\ P_2 \\ P_9 \\ Q_1 \\ \vdots \\ Q_6 \end{bmatrix}$$

The total number of variables in $\vec{z}$ is 9, and T is full column rank. In this case, $k = pm$, given $M\vec{z} = \vec{b}$ ($M = LT$), and since M is square and full rank, $(M)^{-1}$ exists. Therefore, $\vec{z} = (M)^{-1}\vec{b}$, hence reconstruction is possible, and the signal structure of the system can be uniquely identified from the overall structure of Q

3.5 Conclusion

This paper extends the identifiability conditions required for the reconstruction of the signal structure of an LTI system. The notion of the DSF (Q, P), of system was introduced in [24] and the required conditions for reconstruction were also indicated. From previous work, it was shown that if no other information is known about the system, the DSF is identifiable if and only if P is diagonal.

This work shows the necessary and sufficient identifiability conditions for the reconstruction of the DSF given a system's TF matrix G. These results are applicable to the network reconstruction problem even in the cases when the control matrix P is not diagonal. This extension is significant because it identifies when systems that do not necessarily have independent perturbation of measured states are reconstructible.

3.6 Appendix: Existence of Non-Target Specific Systems

Consider the following transfer function:

$$G(s) = \begin{bmatrix} \frac{-(s+1)}{(s+3)(s+5)} & \frac{-2}{(s+5)} \\ \frac{-(s+1)}{(s+2)(s+5)} & \frac{-(s+1)}{(s+2)(s+5)} \end{bmatrix}$$

If we want to determine the corresponding dynamical structure function, we need a priori information about the system. Let's assume that the system has diagonal P (i.e. the system is target specific), applying the process described in Example 8, we get the dynamical structure

function:

$$Q = \begin{bmatrix} 0 & \frac{2s+4}{s+1} \\ \frac{s+3}{s+2} & 0 \end{bmatrix}, \qquad P = \begin{bmatrix} \frac{1}{s+3} & 0 \\ 0 & \frac{1}{s+2} \end{bmatrix}$$

Note that by the definition of the dynamical structure function, Q can not be strictly proper in the sense defined here. Therefore, the assumption that P is diagonal is an incorrect one, there is no dynamical structure function associated with diagonal P for the transfer function given. That means there exist systems that cannot be realized with diagonal P, which motivates the results of this chapter. Note that there are a set of dynamical structure functions associated with a transfer function, this example shows that some systems have sets with no diagonal P.

Chapter 4

Network Semantics of Dynamical Systems

(Published at CDC 2015)

Abstract

Dynamical systems enjoy a rich variety of mathematical representations, from interconnections of convolution operators or rational functions of a complex variable to systems of (possibly stochastic) differential or differential algebraic equations. Although many of these representations can describe the same *behavior*, i.e. represent the same constraints on manifest variables, each one characterizes a different notion of *system structure*. This paper details the differences between the notions of system structure that arise naturally from common system representations, highlighting distinctions between the interconnection of subystems, or *subsystem structure*, and the open-loop causal dependencies among manifest signals, or *signal structure*. We detail many of the subtleties related to signal structures by extending the definition of the dynamical structure function to the general LTI setting.

4.1 Introduction

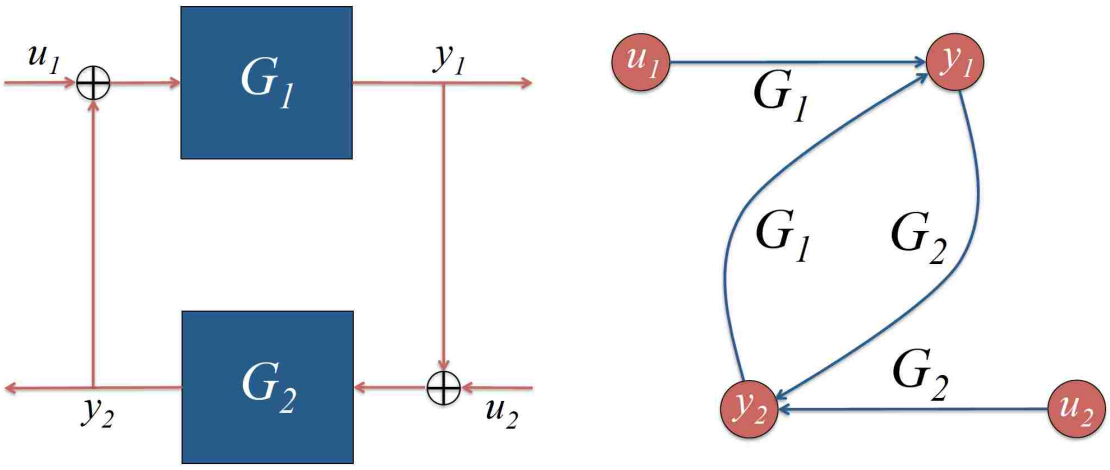
Dynamics and structure are two of the most important properties of a dynamical system. Nevertheless, while a system's dynamics describe its behavior, that is, how it constrains allowed combinations of measured, or *manifest* variables, the notion of structure in a system is a property of its representation. Since a variety of representations can describe the same system as specified by a fixed behavior (e.g. the transfer function matrix of a linear time invariant system as well as any of its state realizations), every system can be associated with a variety of internal structures.

In this work *structure* refers to a directed graph, characterized by a set of nodes or vertices, V, and a set of edges, E, where each edge can be identified with an ordered pair of nodes from V. We may associate both nodes and edges with particular labels, giving the graph its meaning, or *semantics*. The fact, then, that a system may be characterized by multiple structures, each with a distinct interpretation, indicates the existence of a variety of graphs that reveal different insights about the underlying system. This presence of multiple structures associated with any dynamical system raises significant questions about any structural analyses performed on such systems. For example, which type of structure does a network identification algorithm discover, or which type of structure should characterize the design constraints in a distributed or networked control problem? Using the wrong type of structure for different problems can lead to misleading–and even incorrect–results.

4.1.1 Motivating Example

To make these ideas precise, consider the feedback interconnection of two subsystems, G_1 and G_2, as shown in Figure 4.1a, and suppose the dynamics of these subsystems are specified by the following transfer functions:

$$G_1 \;=\; \tfrac{1}{(s+1)(s+2)}, \quad G_2 \;=\; \tfrac{1}{(s+3)(s+4)}. \tag{4.1}$$



(a) *Subsystem structure* is a graph with nodes labeled as systems and edges labeled as signals, typically drawn as a block diagram. It reveals a decomposition of the closed-loop system into an interconnection of subsystems.

(b) *Signal structure* is a graph with nodes labeled as signals and edges labeled as systems, typically drawn as a signal-flow graph. It reveals the open-loop causal dependencies among manifest variables.

Figure 4.1: Two structural views of the same system. Semantic differences in these views emerge from the fact that subsystems in (a) have internal, hidden states that are *necessarily* distinct from those of other subsystems, while edge-systems in (b) may (or may not) share hidden state.

Interconnecting these subsystems yields the closed-loop system, G_{CL}, given by:

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} \frac{G_1}{1-G_1G_2} & \frac{G_1G_2}{1-G_1G_2} \\ \frac{G_2G_1}{1-G_1G_2} & \frac{G_2}{1-G_1G_2} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \tag{4.2}$$

where $Y_1(s)$ indicates the Laplace transform of $y_1(t)$ (similarly for other signals), and the explicit s-dependence is suppressed wherever it should be obvious. Figure 4.1a illustrates the *subsystem structure* of G_{CL}, or the interconnection pattern of subsystems generating the dynamics seen externally as G_{CL}.

Figure 4.1b illustrates a different decomposition of G_{CL}, characterized by the following relationships among the manifest signals:

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} 0 & G_1 \\ G_2 & 0 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} + \begin{bmatrix} G_1 & 0 \\ 0 & G_2 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \tag{4.3}$$

Note that this characterization defines the graphical structure shown in Figure 4.1b and is consistent with G_{CL}, since Equation (4.3) implies:

$$\begin{aligned} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} &= \left(I - \begin{bmatrix} 0 & G_1 \\ G_2 & 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} G_1 & 0 \\ 0 & G_2 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \\ &= \begin{bmatrix} \frac{G_1}{1-G_1G_2} & \frac{G_1G_2}{1-G_1G_2} \\ \frac{G_2G_1}{1-G_1G_2} & \frac{G_2}{1-G_1G_2} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}. \end{aligned} \tag{4.4}$$

The structural decomposition of G_{CL} in Figure 4.1b is called the *signal structure*, since it reveals the (open-loop) dependencies among manifest signals (in contrast to the closed-loop dependencies among manifest signals, which are specified by the entries of G_{CL} itself). These open-loop dependencies characterize a *factorization* of the closed-loop dependencies encoded in the transfer function, G_{CL}, as seen in Equation (4.4). Note that considering nodes (signals) or edges (systems) in the time or frequency domain makes no difference to the analysis, and these labels may thus be interchangeable (Y_1 for y_1, etc.) in the signal structures of LTI systems (likewise for subsystem structures).

While the structural information in Figures 4.1a and 4.1b may appear redundant, the subsystem and signal structures represent very different sets of possible realizations of G_{CL}. The meaning, or *semantics*, of each type of structure derive from these sets of underlying realizations that each structure allows and forbids.

So, for example, knowing the subsystem structure of G_{CL}, as depicted in Figure 4.1a and Equation (4.1), eliminates all realizations that are not fourth order, since both G_1 and G_2 are second order and, by convention, transfer functions are realized minimally. Moreover, this set of fourth order realizations allowed by the subsystem structure can be completely characterized by:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_1 & B_1C_2 \\ B_2C_1 & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$
$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{4.5}$$

where $x_1, x_2 \in \mathbb{R}^2$, and (A_1, B_1, C_1) and (A_2, B_2, C_2) are any minimal realization of G_1 and G_2, respectively.

The signal structure in Figure 4.1b and Equations (4.1) and (4.3), on the other hand, is not as restrictive as the subsystem structure, and thus it is *weaker*, or structurally less informative than the subsystem structure. That is to say, every realization admissible with respect to the subsystem structure, characterized by Equation (4.5), is also allowed by the signal structure–but there are many others as well. For example, the signal structure also admits minimal realizations that do not decompose into two distinct subsystems as shown in Figure 4.1a, such as (A_{CL}, B_{CL}, C_{CL}) given by

$$\begin{bmatrix} -1 & 0 & 1 & 0 \\ 1 & -3 & 0 & 0 \\ 0 & -1 & -2 & 1 \\ 1 & 2 & 0 & -4 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \tag{4.6}$$

as well as non-minimal realizations, such as (A_{CL}, B_{CL}, C_{CL}) given below, found by misinterpreting the signal structure in Figure 4.1b as the interconnection of distinct subsystems:

$$\left[\begin{array}{cccccccc} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -3 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & -4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -3 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4 \end{array}\right], \left[\begin{array}{cc} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{array}\right], \left[\begin{array}{cc} 1 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{array}\right]' \tag{4.7}$$

Each of these have the same signal structure, as shown in Figure 4.1b, but they do not share the subsystem structure in Figure 4.1a. Section 4.3.2 details how to determine the signal structure from a state realization.

The reason for these differences in network semantics is that subsystem structure communicates, by convention, the critical feature that *states internal to one subsystem are not employed by other subsystems.* This state-partitioning property of the subsystem structure restricts the set of admissible realizations much more severely than the signal structure, which remains agnostic as to whether edge-systems share hidden state or not. These semantic differences suggest, for example, that learning signal structure from data often requires much less *a priori* information about a system than learning subsystem structure would, since, for example, learning subsystem structure demands the identification of a *partition* of all system states, including hidden states that are not measured directly. Also, misinterpreting signal structure as subsystem structure can result in technical difficulties for both identification and control since it may lead one to believe that the underlying realization is much higher order than it really is. Consider, for example, Figure 4.1b, interpreting each edge-system as a distinct subsystem, leading to *necessarily* eighth order realizations (a second order subsystem

for each edge-system), as in Equation (4.7). Properly interpreting the signal structure would still admit these eighth order realizations for consideration, but it would also admit all the other realizations discussed above, including all those characterized by Equation (4.5).

The next section explores related works, detailing how different structural analyses in the literature employ different notions of structure. The rest of the paper then focuses on signal structure, arguably the most unfamiliar structural form, providing a detailed description of its properties and relationships to other notions of structure.

4.2 Background and Related Work

Exploring and exploiting system structure is not new, but it continues to be one of the most relevant and exciting topics in systems theory today. Multiple conferences, meetings, workshops, seminars, journals, and other venues focus on cooperative control, multi-agent systems, networked control, network identification, inference over networks, distributed computation, and network science–all of which invite various forms of structural analyses that rely, at some point, on a particular notion of "network" characterizing the structure of a system.

Although there is a rich diversity of contexts and applications for this kind of thinking about structure, we propose that much of the work can be categorized by their use of four distinct notions of system structure. In this section we look at four key structural representations that span the spectrum of informativity of the structure of a system: a system's external or *manifest structure*, the interconnection of subsystems or *subsystem structure*, the open-loop causal dependencies among manifest variables or *signal structure*, and the underlying realization or *complete computational structure* of a system. Although no attempt is made to comprehensively categorize all work dealing with structure, we do demonstrate a number of sources working with each distinct notion of structure.

4.2.1 Manifest Structure

The weakest notion of structure is characterized by a system's external behavior. For single-input single-output systems, this behavioral description contains no structural information about the system and simply specifies its dynamics. Nevertheless, for multi-input multi-output systems, the manifest structure reflects some structural information about the internal closed-loop behavior of the system, where "closed-loop" in this context means the net affect of internal components interacting to create the observed external dynamic behavior. Not all systems are best described in an input-output framework [41], however, and manifest structure is completely consistent with external observations of how the system constrains dependencies among manifest variables. The key feature of this type of structure is that it describes the *net effect* of manifest variables on each other.

Historically a rich conversation and literature has emerged exploring the meaning of how a manifest variable may effect another [20, 39, 40]. Correlation, mutual information, and similar notions have provided rigorous ways to think about these dependencies among manifest variables [19, 53, 61]. Causal dependencies, largely abandoned by work in statistics, have regained momentum through the development of inference methods for Bayesian graphical models and ongoing mechanistic modeling in engineering, physics, and other scientific disciplines. Regardless of how "effect" is measured, however, the important distinction we want to make here is the representation of a system's *closed-loop* or *net effect* among manifest variables.

This type of structural description has a long and distinguished relationship with systems and control theory. Hierarchical, decentralized, and distributed control methods typically characterized these constraints for LTI systems through the sparsity pattern of a transfer function (for frequency domain representations) or convolution model (for time domain representations) [54]. Even recent work continue to use this notion of structure to formulate distributed control problems and subsequent solutions [34, 47, 48, 50, 67]. Nevertheless, researchers have also noted limitations of this structural representation, citing

its inability to adequately describe internal loops and other intricacies of an underlying realization [9, 44, 51, 59]. Clearly other, more detailed representations of structure are necessary for many problems. Nevertheless, here we define manifest structure as follows:

Definition 7. *The **manifest structure**, $\mathcal{M}$, of a system, $\mathcal{S}$, is a weighted directed graph with vertex set $V(\mathcal{M})$ and an edge set $E(\mathcal{M})$ such that:*

- $V(\mathcal{M}) = \{v_1, ..., v_p\}$, *each representing a manifest signal of the system, and*
- $E(\mathcal{M})$ *has an edge from v_i to v_k weighted by the net dynamics relating v_k to v_i,*

where $v \in \mathbb{R}^{p+m}$, are manifest variables. If these manifest variables partition into inputs and outputs, then we expect edges from $u_i,\ i = 1, .., m$ to $y_k,\ k = 1, ..., p$, each labeled with the corresponding net dynamic relationship; for LTI systems these may be the entries of the $p \times m$ transfer function matrix.

4.2.2 Complete Computational Structure

On the other end of the spectrum, the complete computational structure is a specific realization of a system that completely specifies all of its relevant structural and dynamic information. Like a blueprint for building a particular instantiation of the system, each complete computational structure uniquely specifies every other structural representation of the system. Thus, every complete computational structure will generate a unique manifest structure, but each manifest structure will be consistent with an entire set of complete computational structures, or *computational realizations*.

It is tempting and somewhat useful to associate the complete computational structure with a state realization of a dynamic system. Certainly a state realization uniquely defines a system's manifest structure, and every manifest structure has a variety of state realizations to choose from. Nevertheless, sometimes standard state realizations are not sufficient to describe a system's subsystem structure, and certain classes of differential algebraic equations become the desired model class characterizing the complete computational structure. In any event,

given a particular state realization:

$$\begin{aligned} \dot{x} &= f(x, u, \psi), \\ y &= g(x, u, \psi) \end{aligned} \tag{4.8}$$

with state variables $x \in \mathbb{R}^n$, controlled inputs $u \in \mathbb{R}^m$, uncontrolled inputs $\psi \in \mathbb{R}^l$, and measured outputs $y \in \mathbb{R}^p$, the complete computational structure is a graph characterized as follows:

Definition 8. *The **complete computational structure**, $\mathcal{C}$, of a system, $\mathcal{S}$, is a weighted directed graph with vertex set $V(\mathcal{C})$ and an edge set $E(\mathcal{C})$ such that:*

- $V(\mathcal{C}) = \{u_1, ..., u_m, \psi_1, ..., \psi_l, x_1, ..., x_n, y_1, ..., y_p\}$, *each representing signals of the system, and*
- $E(\mathcal{C})$ *has an edge from u_i, ψ_j, y_k, or x_v to y_w or x_z weighted by the relevant dependencies of y_w or x_z on u_i, ψ_j, y_k, or x_v, respectively.*

Note that the set of elements in ψ can affect x's and y's independently, this allows for a model of both process noise and measurement noise.

There are many subtle technicalities associated with specifying either a state realization or an associated complete computational structure of a system, generally dealing with the existence and uniqueness of solutions to the resulting equations. Nevertheless, characterizing the set of computational realizations of a specific dynamic system helps quantify the information cost for recovering structural information from data, interpreting the meaning of partial structure representations, and understanding equivalences between different system representations.

In fact, many network identification algorithms are formulated in terms of recovering a state realization from input-output data [21]. In [38] it is shown that one must essentially have full state measurements (or equivalent information) to recover a particular state realization of an LTI system from data, and [58] uses this to reconstruct biochemical reaction networks.

Others attempt to circumvent the need for full state measurements through other assumptions, such as parsimonious state interactions, [13, 22, 66], although other work criticises these assumptions for biological systems, [68]. In the end, identifying the complete computational structure from data is difficult and generally prohibitively expensive, leading to heuristic searches for realizations that hopefully correspond favorably to the one that generates the data used for identification, [12].

State representations have long been the work horse for a number of computational solutions to important problems in identification and control. Recent work includes explorations of sparse measurement and actuation policies in network control schemes. For example, in [43] the authors develop an efficient algorithm for sensor placement to detect and isolate any possible link failure using a small number of sensors.

4.2.3 Subsystem Structure

The interconnection of subsystems is the typical notion of structure used for cooperative control and multi-agent systems [33, 37, 46], as well as a variety of other areas [8].

The key property of subsystem structure that gives it its meaning is the fact that *states internal to one subsystem are not employed by other subsystems.* This implies that the subsystem structure partitions the state of the interconnected system, and searches for the correct subsystem structure for an unknown interconnected system involve a search over possible partitions, which can be computationally intractable. On the other hand, knowing the correct subsystem structure of a system conveys a lot of information about the underlying complete computational structure. Here we define the subsystem structure as follows:

Definition 9. *The* ***subsystem structure****, $\mathcal{R}$, of a system, $\mathcal{S}$, is a weighted directed graph with vertex set $V(\mathcal{R})$ and an edge set $E(\mathcal{R})$ such that:*

- $V(\mathcal{R}) = \{u_1, ..., u_m, \psi_1, ..., \psi_l, S_1, ..., S_q, y_1, ..., y_p\}$*, each representing a manifest signal,* u*,* ψ *and* y*, or subsystem,* S*, of the system, and*
- $E(\mathcal{R})$ *has an edge from*

- *u_i or ψ_j to S_k if S_k is dependent on u_i or ψ_j, respectively,*
- *S_v to S_w if S_w is dependent on S_v,*
- *S_z to y_a if y_a is dependent on S_z.*
- *Each edge is labeled with the associated signal variable.*

4.2.4 Signal Structure

The signal structure is a partial structure representation of a system that characterizes the open-loop causal dependencies among manifest variables. That is to say, it describes the dynamics along paths in the complete computational structure from one manifest variable to another, possibly transversing hidden states. For LTI systems, the signal structure is consistent with a left co-prime factorization [51] of the transfer function matrix, and various system representations generate a signal structure, including the dynamical structure function [24] and linear dynamical graphs [36].

A variety of researchers working on network identification adopt a formulation that leads to identifying the signal structure of a system [7, 16, 27, 36, 58, 65, 73], although sometimes it is not clear in the work whether the authors distinguish the semantic differences between signal and subsystem structure.

The key nuance about signal structure that distinguishes its semantics from subsystem structure is the fact that component systems, or "edge systems", do not need to necessarily partition system state; edge systems may "share" hidden state. That is to say, the set of underlying computational realizations consistent with a particular signal structure is very different from the set of realizations that would treat each edge system as a distinct, independent subsystem.

In [74], the authors link dynamical structure functions to other signal structure techniques utilizing Granger causality. One such technique uses a representation known as the directed information graph, [42], which linked networks of stochastic processes to Granger causality. Another inference technique, developed in [36], uses linear dynamical graphs, a

signal structure representation based on Granger causality, that is similar to the dynamical structure function.

Aside from network inference, signal structure has also been useful in the development of a procedure for designing a stabilizing decentralized controller, [44]. In other work, [17, 45], the signal structure was used to model the attack surface of a system to single and multiple link attacks. The signal structure is defined as follows:

Definition 10. *The **signal structure**, $\mathcal{W}$, of a system, $\mathcal{S}$, is a weighted directed graph with vertex set $V(\mathcal{W})$ and an edge set $E(\mathcal{W})$ such that:*

- $V(\mathcal{C}) = \{u_1, ..., u_m, \psi_1, ..., \psi_l, y_1, ..., y_p\}$, *each representing a manifest signal of the system, and*
- $E(\mathcal{C})$ *has an edge from u_i, ψ_j, y_k, to y_v if y_v is dependent on u_i, ψ_j, y_k, respectively.*

Each edge of the signal structure is weighted with a dynamic operator characterizing the dynamics between the associated variables if all other manifest variables are set to zero. Details about these notions are described in the next section.

4.3 Partial Structure Representations

In this section, we focus on detailing the mathematical descriptions of two partial structure representations of the system, the subsystem structure and the signal structure, since both the transfer function (manifest structure) and state space representations (complete computational structure) of systems are well-studied in the existing literature.

4.3.1 Subsystem Structure

Definition 11. *The linear **subsystem structure** dynamics is for the i^{th} subsystem is given by*

$$\dot{x}_i = A_i x_i + \sum_{j=1}^{n} D_{ij} x_j + B_i u_i \tag{4.9}$$

where $i = 1, ..., q$, $j \neq i$, $x_k \in \mathbb{R}^{n_k}$, $u_k \in \mathbb{R}^{m_k}$, *and* A_i, D_{ij}, *and* B_i *are of appropriate dimension.*

Note that subsystems only interact with other subsystems as established through the subsystem structure. However, certain problems might couple the subsystems through their objectives.

Moreover, an implicit part of the definition of subsystem structure is that for a state $r \in x_i \implies r \notin x_j$ for any $j \neq i$. This means that states are not shared between subsystems and, more importantly, there are no shared hidden states in the subsystem structure. This characteristic that distinguishes it from the signal structure of a system.

4.3.2 Signal Structure

The signal structure is characterized by an equation of the form:

$$Y = QY + PU. \tag{4.10}$$

Although a number of researchers develop system representations in this form, we will use the dynamical structure function as a vehicle for understanding the properties of this structural representation.

The dynamical structure function of a system was originally defined in [24]. This work seeks to make three critical extensions to the original definition, given a state space of the form

$$\begin{aligned} \dot{x} &= Ax + Bu + F\psi \\ y &= Cx + Du + H\psi \end{aligned} \tag{4.11}$$

where A, B, C, D, F and H are matrices of appropriate dimension:

1. to include noisy perturbations, ψ, which allows for the separation of perturbations inflicted by an external source, like an attacker or intrinsic noise, from controlled inputs by the user,

2. to allow for causal definitions of systems, which allows for stochastic processes to be modelled, and

3. to define a transformation of the system that is invariant to state permutations, since the dynamical structure function should not change if the states are simply renumbered.

Definition 12. *Consider a subsystem structure where each subsystem, S_i, has a state space representation of the form given in (4.11). The interconnections between structures are causal. The procedure for calculating the causal* ***dynamical structure function*** *is as follows:*

1. *For each subsystem, use the follows steps to determine the strictly causal dynamical structure function for the i^{th} subsystem:*

   (a) *Let p be the rank of C, and assume without loss of generality that the outputs $y = [y_1' \;\; y_2']'$, $y_1 \in \mathbb{R}^p$ and $y_2 \in \mathbb{R}^{(l-p)}$, are ordered so the first p rows of C are linearly independent, i.e.*

$$C = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}$$

   with $C_1 \in \mathbb{R}^{p \times n}$ being full row rank. The dynamical structure function of the subsystem of the form (4.11) with respect to y_1 is then given by a pair of $(l \times p)$ and $(l \times m)$ real rational matrix functions, $(\hat{Q}(s), \hat{P}(s))$, defined over the Laplace variable, $s \in \mathbb{C}$, and constructed by the following procedure:

   (b) *Create the $(n \times n)$ state transformation:*

$$T = \begin{bmatrix} C_1' & E_1 \end{bmatrix}', \tag{4.12}$$

   where $E_1 \in \mathbb{R}^{n \times (n-p)}$ is any basis of the null space of C_1, with

$$T^{-1} = \begin{bmatrix} R_1 & E_1 \end{bmatrix}, \tag{4.13}$$

   where $R_1 = C_1'(C_1 C_1')^{-1}$.

(c) *Change basis such that* $z = Tx$, *yielding* $\hat{A} = TAT^{-1}$, $\hat{B} = TB$, $\hat{C} = CT^{-1}$, $\hat{D} = D$, $\hat{F} = TF$, *and* $\hat{H} = H$, *and partitioned commensurate with the block partitioning of* T *and* T^{-1} *to give*

$$\begin{aligned}\begin{bmatrix}\dot{z}_1\\ \dot{z}_2\end{bmatrix} &= \begin{bmatrix}\hat{A}_{11} & \hat{A}_{12}\\ \hat{A}_{21} & \hat{A}_{22}\end{bmatrix}\begin{bmatrix}z_1\\ z_2\end{bmatrix} + \begin{bmatrix}\hat{B}_1\\ \hat{B}_2\end{bmatrix}u + \begin{bmatrix}\hat{F}_1\\ \hat{F}_2\end{bmatrix}\psi\\ \begin{bmatrix}y_1\\ y_2\end{bmatrix} &= \begin{bmatrix}I & 0\\ \hat{C}_{21} & 0\end{bmatrix}\begin{bmatrix}z_1\\ z_2\end{bmatrix} + \begin{bmatrix}\hat{D}_1\\ \hat{D}_2\end{bmatrix}u + \begin{bmatrix}\hat{H}_1\\ \hat{H}_2\end{bmatrix}\psi\end{aligned} \tag{4.14}$$

Note that while it is easily seen that $C_1R_1 = I$ *and* $C_1E_1 = 0$, *but the fact that* $C_2E_1 = 0$ *may demand some reflection. The reason this is true is because every row of* C_2 *is in the row space of* C_1. *If it were not so, then either the rank of* C *would be greater than* p *or* C_1 *would not be composed of* p *linearly independent rows. Being in the row space of* C_1, *each row in* C_2 *is thus also orthogonal to every vector in* E_1, *which spans the orthogonal complement of the row space of* C_1.

(d) *Assume zero initial conditions, take Laplace transforms, and solve for* Z_2, *yielding*

$$\begin{aligned}sZ_1 &= \left[\hat{A}_{11} + \hat{A}_{12}(sI - \hat{A}_{22})^{-1}\hat{A}_{21}\right]Z_1 + \left[\hat{B}_1 + \hat{A}_{12}(sI - \hat{A}_{22})^{-1}\hat{B}_2\right]U +\\ &\quad \left[\hat{F}_1 + \hat{A}_{12}(sI - \hat{A}_{22})^{-1}\hat{F}_2\right]\Psi\end{aligned}$$

$$\begin{bmatrix}Y_1\\ Y_2\end{bmatrix} = \begin{bmatrix}I & 0\\ \hat{C}_{21} & 0\end{bmatrix}\begin{bmatrix}Z_1\\ Z_2\end{bmatrix} + \begin{bmatrix}\hat{D}_1\\ \hat{D}_2\end{bmatrix}U + \begin{bmatrix}\hat{H}_1\\ \hat{H}_2\end{bmatrix}\Psi \tag{4.15}$$

where Z, U, Y, *and* Ψ *denote the Laplace transforms of* z, u, y, *and* ψ *respectively.*

(e) *For notational simplicity, define:*

$$W(s) = \hat{A}_{11} + \hat{A}_{12}(sI - \hat{A}_{22})^{-1}\hat{A}_{21} \tag{4.16}$$

$$V(s) = \hat{B}_1 + \hat{A}_{12}(sI - \hat{A}_{22})^{-1}\hat{B}_2 \tag{4.17}$$

$$L(s) = \hat{F}_1 + \hat{A}_{12}(sI - \hat{A}_{22})^{-1}\hat{F}_2 \tag{4.18}$$

and let $D_W(s) = \text{diag}(W(s))$ *be a diagonal matrix function composed of the diagonal entries of* $W(s)$.

(f) Define $Q(s) = (sI - D_W)^{-1}(W - D_W)$, $P(s) = (sI - D_W)^{-1}V$, *and* $R(s) = (sI - D_W)^{-1}L$ *yielding*

$$\begin{aligned} Z_1 &= Q(s)Z_1 + P(s)U + R(s)\Psi \\ \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} &= \begin{bmatrix} I & 0 \\ \hat{C}_{21} & 0 \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} + \begin{bmatrix} \hat{D}_1 \\ \hat{D}_2 \end{bmatrix} U + \begin{bmatrix} \hat{H}_1 \\ \hat{H}_2 \end{bmatrix} \Psi \end{aligned} \tag{4.19}$$

(g) Noting from (4.19) that $Z_1 = Y_1 - \hat{D}_1 U - \hat{H}_1\Psi$, *the dynamical structure function of (4.11) with respect to* y_1 *is then given by:*

$$\hat{Q}(s) = \begin{bmatrix} Q(s) \\ C_{21} \end{bmatrix},$$

$$\hat{P}(s) = \begin{bmatrix} P(s) + (I - Q(s))\hat{D}_1 \\ \hat{D}_2 - \hat{C}_{21}\hat{D}_1 \end{bmatrix}, \tag{4.20}$$

$$\hat{R}(s) = \begin{bmatrix} R(s) + (I - Q(s))\hat{H}_1 \\ \hat{H}_2 - \hat{C}_{21}\hat{H}_1 \end{bmatrix}$$

which satisfies

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \hat{Q}(s)Y_1 + \hat{P}(s)U + \hat{R}(s)\Psi \tag{4.21}$$

2. *Given the dynamical structure functions* $(\hat{Q}_i(s), \hat{P}_i(s), \hat{R}_i(s))$ *for each subsystem* S_i*, interconnect them using block diagram algebra to get the dynamical structure function of the overall system,* $(\tilde{Q}(s), \tilde{P}(s), \tilde{R}(s))$.

Note that systems of the form

$$\begin{aligned} \dot{x} &= \hat{A}x + \hat{B}u \\ y &= \begin{bmatrix} I & 0 \end{bmatrix} x \end{aligned} \tag{4.22}$$

generate transformations of the form $T = T^{-1} = I$ and have $y = y_1$ (y_2 is empty), thereby leading to $\hat{Q} = Q$ and $\hat{P} = P$, reducing to the definition of the dynamical structure function from [24]. In this way, this definition of the dynamical structure function is a natural generalization extending the earlier theory.

4.4 Properties of the Extended Dynamical Structure Function

In this section, we detail the properties of the transformation on any state space system representation to put it in the form (4.14) from which the strictly causal dynamical structure function is defined. Extending these properties to the causal case is future work.

Lemma 5. (Invariance to a Class of Block Diagonal Transformations) *Given a system* (A, B, C, D, F, H) *of the form (4.14) with dynamical structure function* $(\hat{Q}, \hat{P}, \hat{R})$*, then* $(\hat{Q}, \hat{P}, \hat{R})$ *is invariant to block diagonal state transformations; that is, the set of systems characterized by block diagonal state transformations,*

$$\mathcal{S} = \{(MAM^{-1}, MB, CM^{-1}, D, MF, H) \mid M = \begin{bmatrix} I_{p\times p} & 0 \\ 0 & M_{22} \end{bmatrix}\},$$

with M_{22} any invertible matrix of appropriate size, all share the same dynamical structure function, $(\hat{Q}, \hat{P}, \hat{R})$.

Proof. Transforming the given system, $\bar{z} = Mz$, yields

$$\begin{aligned}
\begin{bmatrix} \dot{\bar{z}}_1 \\ \dot{\bar{z}}_2 \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12}M_{22}^{-1} \\ M_{22}A_{21} & M_{22}A_{22}M_{22}^{-1} \end{bmatrix} \begin{bmatrix} \bar{z}_1 \\ \bar{z}_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ M_{22}B_2 \end{bmatrix} u + \begin{bmatrix} F_1 \\ M_{22}F_2 \end{bmatrix} \psi \\
\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} &= \begin{bmatrix} I & 0 \\ C_{21} & 0 \end{bmatrix} \begin{bmatrix} \bar{z}_1 \\ \bar{z}_2 \end{bmatrix} + \begin{bmatrix} D_1 \\ D_2 \end{bmatrix} u + \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} \Psi
\end{aligned} \tag{4.23}$$

which leads to

$$\begin{aligned}
W(s) &= A_{11} + A_{12}M_{22}^{-1}(sI - M_{22}A_{22}M_{22}^{-1})^{-1}M_{22}A_{21} \\
&= A_{11} + A_{12}(sI - A_{22})^{-1}A_{21},
\end{aligned} \tag{4.24}$$

$$\begin{aligned}
V(s) &= B_1 + A_{12}M_{22}^{-1}(sI - M_{22}A_{22}M_{22}^{-1})^{-1}M_{22}B_2 \\
&= B_1 + A_{12}(sI - A_{22})^{-1}B_2.
\end{aligned} \tag{4.25}$$

$$\begin{aligned}
L(s) &= F_1 + A_{12}M_{22}^{-1}(sI - M_{22}A_{22}M_{22}^{-1})^{-1}M_{22}F_2 \\
&= F_1 + A_{12}(SI - A_{22})^{-1}F_2.
\end{aligned} \tag{4.26}$$

Since $W(s)$, $V(s)$, and $L(s)$ are invariant to M_{22}, $(\hat{Q}, \hat{P}, \hat{R})$ also remain unchanged with respect to M_{22}. □

Lemma 5 shows that the dynamical structure function is invariant to transformations on the hidden states provided they only involve other hidden states.

Theorem 7. (Invariance to Basis of the Null Space) *Given a system (A, B, C, D, F, H) as in (4.11), consider two distinct bases of the null space of C, $E \neq \bar{E}$, with corresponding*

state transformations:

$$T = \begin{bmatrix} C_1 \\ E' \end{bmatrix}, \quad \bar{T} = \begin{bmatrix} C_1 \\ \bar{E}' \end{bmatrix},$$

as in (4.12), and each leading to its corresponding dynamical structure function, $(\hat{Q}, \hat{P}, \hat{R})$ *and* $(\bar{Q}, \bar{P}, \bar{R})$ *as in (4.20). Then* $(\hat{Q}, \hat{P}, \hat{R}) = (\bar{Q}, \bar{P}, \bar{R})$.

Proof. Let $z = Tx$ and $\bar{z} = \bar{T}x$. Then $\bar{z} = \bar{T}T^{-1}z$:

$$\bar{T}T^{-1} = \begin{bmatrix} C_1 \\ \bar{E}' \end{bmatrix} \begin{bmatrix} R_1 & E \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & \bar{E}'E \end{bmatrix} \tag{4.27}$$

where $R_1 = C_1'(C_1C_1')^{-1}$. The block diagonal structure of $\bar{T}T^{-1}$ then ensures, by Lemma 5, that the dynamical structure function produced for $\bar{z}$ is the same as that for z, i.e. $(\hat{Q}, \hat{P}, \hat{R}) = (\bar{Q}, \bar{P}, \bar{R})$. □

Theorem 8. (Invariance to State Permutations) *Consider a system as in (4.11) with state matrices* (A, B, C, D, F, H) *and dynamical structure function* $(\hat{Q}, \hat{P}, \hat{R})$. *Then* $(\hat{Q}, \hat{P}, \hat{R})$ *is invariant to state permutations; that is, the set of systems characterized by state permutations,*

$$\mathcal{S} \;=\; \{(MAM^{-1}, MB, CM^{-1}, D, MF, H) \mid M \text{ is a permutation matrix}\},$$

all share the same dynamical structure function, $(\hat{Q}, \hat{P}, \hat{R})$, *up to a permutation.*

Proof. The result follows from the fact that that the state transformation selected in Step 1 of constructing the dynamical structure function transforms each system in the set $\mathcal{S}$ to the same system for Step 2; the resulting dynamical structure function is thus the same, up to a permutation. To see this, consider the transformation T constructed for the unpermuted

system, (A, B, C, D, F, H):

$$T = \begin{bmatrix} C_1 \\ E_1' \end{bmatrix}, \quad T^{-1} = \begin{bmatrix} R_1 & E_1 \end{bmatrix},$$

and compare with the transformation T_M constructed for any permuted system, $(MAM^{-1}, MB, CM^{-1}, D, MF, H)$:

$$T_M = \begin{bmatrix} C_1 \\ E_1' \end{bmatrix} M^{-1} = TM^{-1},$$
$$T_M^{-1} = M \begin{bmatrix} R_1 & E_1 \end{bmatrix} = MT^{-1}.$$

Applying each set of transformations to their respective systems yields the same transformed system, thus producing the same dynamical structure function:

$$\begin{aligned}(T_M MAM^{-1}T_M^{-1}, T_M MB, CM^{-1}T_M^{-1}, D, T_M MF, H) \\ = (TAT^{-1}, TB, CT^{-1}, D, TF, H).\end{aligned}$$

□

These results clearly demonstrate that the dynamical structure function is well defined for any state realization or interconnection of subsystems characterized by state realizations. Moreover, the associated signal structure is a meaningful description of the open-loop causal dependencies among manifest variables, even when these dependencies are not strictly causal. The semantics of this notion of structure then derives from the set of realizations consistent with a given dynamical structure function, and they can be quite different from those associated with a comparable signal structure, as illustrated in the motivating example in Section 1.

4.5 Conclusion

Dynamical systems have a variety of representations, each of which may describe the same dynamic behavior but communicate different notions of structure. A typical example for LTI systems would be a transfer function (or, equivalently, a convolution model), along with an associated state realization.

Partial structure representations, such as cooperative multiagent models, linear dynamical graphs, directed information graphs, and dynamical structure functions, characterize a subset of realizations consistent with a given input-output model. This subset gives each notion of structure its meaning, or semantics.

In this paper we categorize structural representations into two main groups, subsystem and signal structures, depending on whether component systems necessarily isolate internal state from other component subsystems. We then catalog a number of technical results that compare and contrast these different notions of structure.

Chapter 5

Semantics and Identifiabilty of the Structured Linear Fractional Transformation

In this chapter we present three methods for comparing partial structure representations (the dynamical structure function and the structured linear fractional transformation) of a system: order of minimal realizations, sets of network semantics and information cost for network reconstruction.

5.1 Orders of Minimal Realizations

In order to develop orders of minimal realizations, we first introduce the following concepts:

1. controllability and observability of a state space model,
2. the Kalman decomposition of a state space model,
3. the order of a state space model,
4. the order of minimal realizations, and
5. the relationships among the respective orders of the minimal realizations of system representations.

Remember that a zero intricacy state space model of a system is a set of equations of the form:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{5.1}$$

5.1.1 Controllability and Observability of a State Space Model

The observability of a state space model details whether the initial state can be learned from a given input and output, while the controllability of a state space model describes whether a control input can be designed to steer the system to the origin from an arbitrary state [30]. Formal definitions of controllability and observability of a state space model are given in Definitions 13 and 14, respectively.

Definition 13. *The pair (A, B) of a state space system of the form (5.1) is* controllable *if every state can be transferred to the origin in finite time.*

Definition 14. *The pair (A, C) of a state space system of the form (5.1) is* observable *if every state at time $t_0 \geq 0$ can be determined from future inputs and outputs $u(t)$ and $y(t)$, for $t \in [t_0, t_1]$.*

5.1.2 Kalman Decomposition of a State Space Realization

Given the definitions of observability and controllability, the *Kalman decomposition* of a state space model of the form (5.1) is given in Definition 15.

Definition 15. *For every state space model of the form (5.1), there exists a similarity transformation that takes it to the form of a Kalman decomposition, which is:*

$$\begin{bmatrix} \dot{x}_{co}(t) \\ \dot{x}_{\bar{c}o}(t) \\ \dot{x}_{c\bar{o}}(t) \\ \dot{x}_{\bar{c}\bar{o}}(t) \end{bmatrix} = \begin{bmatrix} A_{co} & A_{\times o} & 0 & 0 \\ 0 & A_{\bar{c}o} & 0 & 0 \\ A_{c\times} & A_{\times\times} & A_{c\bar{o}} & A_{\times\bar{o}} \\ 0 & A_{\bar{c}\times} & 0 & A_{\bar{c}\bar{o}} \end{bmatrix} \begin{bmatrix} x_{co}(t) \\ x_{\bar{c}o}(t) \\ x_{c\bar{o}}(t) \\ x_{\bar{c}\bar{o}}(t) \end{bmatrix} + \begin{bmatrix} B_{co} \\ 0 \\ B_{c\bar{o}} \\ 0 \end{bmatrix} u(t) \tag{5.2}$$

$$y = \begin{bmatrix} C_{co} & C_{\bar{c}o} & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{co}(t) \\ x_{\bar{c}o}(t) \\ x_{c\bar{o}}(t) \\ x_{\bar{c}\bar{o}}(t) \end{bmatrix} + Du(t)$$

where x_c are the controllable states, $x_{\bar{c}}$ are the uncontrollable states, x_o are the observable states, and $x_{\bar{o}}$ are the unobservable states, e.g. x_{co} is observable and controllable, while $x_{\bar{c}\bar{o}}$ is unobservable and uncontrollable.

5.1.3 Orders of State Space Realizations

The order of a state space realization is detailed in Definition 16.

Definition 16. *Given a state space model of the form (5.1), the* order *of the system is the number of states in the system, n.*

Instances of a system representation such as the transfer function, the structured linear fractional transformation, or the dynamical structure function are associated with a set of state space models of various orders. Thus, we introduce the notion of a minimal realization of a system in Definition 17.

Definition 17. *The* order of a minimal realization *of a state space model with a given*

1. *transfer function,*
2. *structured linear fractional transformation, or*
3. *dynamical structure function*

is the smallest number of states, $\hat{n}$, in an associated state space model that preserves the structure and dynamics of the original representation.

The order of a minimal realization depends on the representation from which it is being realized. The names of the orders of each minimal realization are described in Definitions 18, 19, and 20.

Definition 18. *The* McMillan degree *of a system is the order of any minimal realization of a transfer function $G(s)$.*

Definition 19. *The* structural degree *of a system is the order of any minimal realization of a dynamical structure function $(Q(s), P(s))$.*

Definition 20. *The* subsystem degree *of a system is the order of any minimal realization of a structured linear fractional transformation* $(N, S(s))$.

Given the definitions of a minimal realization and the associated names, we now compare orders of minimal realizations for each system representation.

Order of Minimal Realizations of a Transfer Function

In order to detail the order of a minimal realization of a transfer function, we include several well known results in Lemmas 6, 7, and 8.

Lemma 6. *A state space model of the form (5.1) is controllable if and only if* $rank(\mathscr{C}) = n$, *where*

$$\mathscr{C} = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix} \tag{5.3}$$

is the controllability matrix of the system and $rank(\mathscr{M})$ *is the number of linearly independent rows or columns of* $\mathscr{M}$.

Lemma 7. *A state space model of the form (5.1) is observable if and only if* $rank(\mathscr{O}) = n$, *where*

$$\mathscr{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{5.4}$$

is the observability matrix of the system.

Lemmas 6 and 7 are essentially showing that the rank of the controllability and observability matrices equal the number of controllable and observable states in the system, respectively.

Lemma 8. *The controllability and observability of a system cannot be changed by a state transformation.*

Proof. Consider a state transformation of the form $z(t) = Tx(t)$ on a system of the form (5.1) where T is any invertible matrix. The controllability matrix of the transformed system is:

$$\begin{aligned}
\hat{\mathscr{C}} &= \begin{bmatrix} TB & TAT^{-1}TB & \dots & (TAT^{-1})^{n-1}TB \end{bmatrix} \\
&= \begin{bmatrix} TB & TAT^{-1}TB & \dots & TA^{n-1}T^{-1}TB \end{bmatrix} \\
&= \begin{bmatrix} TB & TAB & \dots & TA^{n-1}B \end{bmatrix} \\
&= T\begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix} \\
&= T\mathscr{C}
\end{aligned} \tag{5.5}$$

Since T is invertible, $rank(\hat{\mathscr{C}}) = rank(\mathscr{C})$, so the transformed system has the same number of controllable states as the original system. Similarly, $\hat{\mathscr{O}} = \mathscr{O}T^{-1} \implies rank(\hat{\mathscr{O}}) = rank(\mathscr{O})$, completing the proof. □

Given the preliminary results, the order or *degree* of a minimal realization of a transfer function is shown in Lemma 9.

Lemma 9. *The McMillan degree of a realization of a transfer function is the number of states that are both observable and controllable in a system, denoted* n_{co}.

Proof. Given a system with a Kalman decomposition as given in (5.2), the associated transfer function is:

$$\begin{bmatrix} C_{co} & C_{\bar{c}o} & 0 & 0 \end{bmatrix} \left(\begin{bmatrix} sI - A_{co} & -A_{\times o} & 0 & 0 \\ 0 & sI - A_{\bar{c}o} & 0 & 0 \\ -A_{c\times} & -A_{\times\times} & sI - A_{c\bar{o}} & -A_{\times\bar{o}} \\ 0 & -A_{\bar{c}\times} & 0 & sI - A_{\bar{c}\bar{o}} \end{bmatrix} \right)^{-1} \begin{bmatrix} B_{co} \\ 0 \\ B_{c\bar{o}} \\ 0 \end{bmatrix} + D$$

using the Sherman-Morrison-Woodbury formula we get

$$\begin{bmatrix} C_{co} & C_{\bar{c}o} & 0 & 0 \end{bmatrix} \begin{bmatrix} (sI - A_{co})^{-1} & (sI - A_{co})^{-1} A_{\times o} (sI - A_{\bar{c}o})^{-1} & 0 & 0 \\ 0 & (sI - A_{\bar{c}o})^{-1} & 0 & 0 \\ * & * & * & * \\ 0 & * & 0 & * \end{bmatrix} \begin{bmatrix} B_{co} \\ 0 \\ B_{c\bar{o}} \\ 0 \end{bmatrix} + D$$

which evaluates to

$$C_{co}(sI - A_{co})^{-1} B_{co} + D$$

The poles of the transfer function are the poles of $(sI - A_{co})^{-1}$, by Cramer's rule the denominator polynomial is $det(sI - A_{co})$, where $det(M)$ is defined as the determinant of a square matrix M. This polynomial is the characteristic polynomial of A_{co}, whose roots are the eigenvalues of A_{co}. This means that the order of any minimal realization of a transfer function (i.e. the McMilan degree) is equal to the number of observable and controllable states in the system. □

Order of Minimal Realizations of a Dynamical Structure Function

Unlike the transfer function, determining the minimal order of a realization from a given dynamical structure function, i.e. the structural degree, using notions of observability and controllability is still an open problem. However, there is enough known about the issue for it to be useful here, including:

1. The order n_{co} of a G minimal realization is a **lower bound** on the order of a realization consistent with $(Q(s), P(s))$ [72].
2. Any minimal realization of the intermediate representation $(W(s), V(s))$ associated with a given dynamical structure function $(Q(s), P(s))$ is observable. [72]

We denote the structural degree as n_{dsf} and we know the relationship between the minimal order of the transfer function and the minimal order of the dynamical structure function is

$n_{co} \leq n_{dsf}$. This can be written as

$$\text{McMillan degree} \leq \text{structural degree} \tag{5.6}$$

Order of Minimal Realizations of a Structured Linear Fractional Transformation

In order to compare the degrees of the structured linear fractional transformation and the dynamical structure function, we must first understand a key idea that distinguishes them: the notion of shared hidden state. The two partial structure system representations are different for many reasons, one of which is because distinct links in the dynamical structure function can contain the same state, while this is not allowable in the structured linear fractional transformation of a system. In Example 10, a simple system containing a shared hidden state is detailed to demonstrate this concept.

Example 10. *Consider the following state space model:*

$$\begin{aligned}
\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \end{bmatrix} &= \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 1 \\ 0 & 0 & -3 & 1 \\ 1 & 0 & 0 & -4 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \end{bmatrix} \\
\begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix}
\end{aligned} \tag{5.7}$$

The associated computational structure of (5.7) is shown in Figure 5.1. The associated

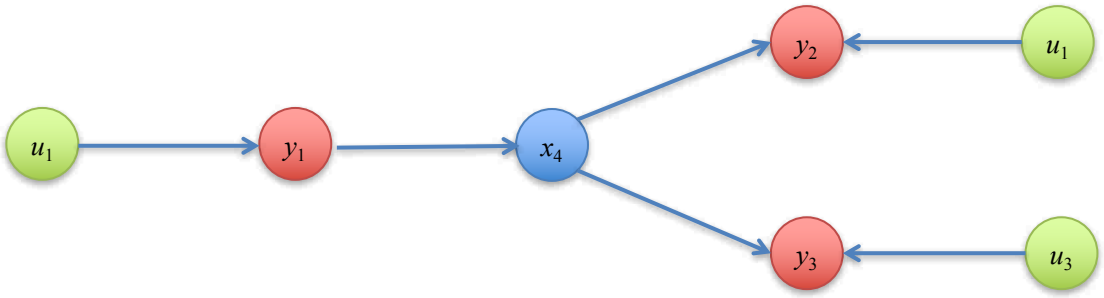


Figure 5.1: Computational structure of the state space model in (5.7). The red nodes represent the measured states of the system, the blue node represents the hidden nodes of the system, and the green nodes are external inputs into the system.

dynamical structure function is:

$$\hat{Q}(s) = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{s^2+6s+8} & 0 & 0 \\ \frac{1}{s^2+7s+12} & 0 & 0 \end{bmatrix} \qquad \hat{P}(s) = \begin{bmatrix} \frac{1}{s+1} & 0 & 0 \\ 0 & \frac{1}{s+2} & 0 \\ 0 & 0 & \frac{1}{s+3} \end{bmatrix} \tag{5.8}$$

Note that the fourth state in (5.7) appears in both the transfer function $Q_{21}(s)$ and $Q_{31}(s)$ since it links the first state to the second state ***and*** *the first state to the third state, as shown in Figure 5.2.*

One final definition we require before detailing the relationship of the subsystem degrees to the structural and McMillan degrees is to define the subsystem structure as a graphical dual of the signal structure.

Definition 21. *The subsystem structure as a* ***graphical dual*** *of the signal structure is defined as follows:*

1. *each $v_j \in V(\mathscr{W})$ becomes an element $v_j \in V(\mathscr{S})$ for all j,*
2. *each $e_i \in E(\mathscr{W})$ becomes an element $S_i \in V(\mathscr{S})$, for $i = 1, ..., p^2 - p + pm$, and*

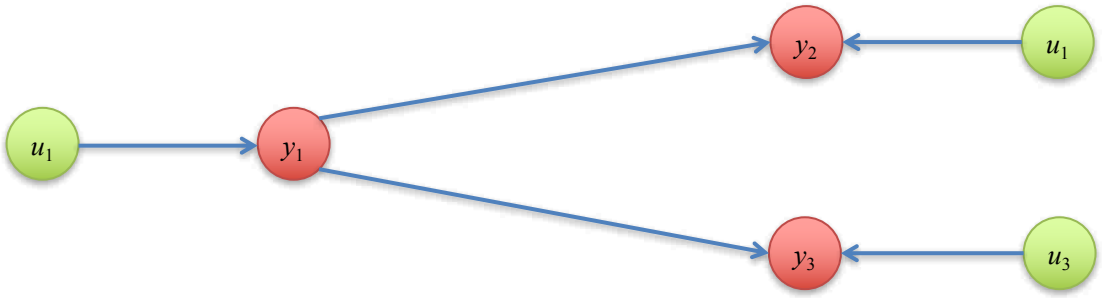


Figure 5.2: Signal structure of the dynamical structure function in (5.8). The red nodes are the measured states in the system and the green nodes are external inputs into the system.

3. *each* $e_i \in E(\mathscr{W})$ *becomes two edges* $e_j, e_h \in E(\mathscr{S})$*, where* e_j *is from the same input or ouput from the signal structure to the subsystem created above and* e_h *is from the subsystem created above to the same output from the signal structure*

With the definition of shared hidden state and graphical dual, we are now equipped to detail the relationships between orders of minimal realizations between the transfer function, dynamical structure function, and structured linear fractional transformation.

Lemma 10. *Given a dynamical structure function,* $(Q(s), P(s))$*, with structural degree* n_{dsf} *and the structured linear fractional transformation which is its graphical dual,* $(N, S(s))$*, with subsystem degree,* n_{sub}*, then* $n_{dsf} \leq n_{sub}$

Proof. The proof can be split into two cases:

1. **Case 1: No Shared Hidden States** Each non-zero link in the dynamical structure function, i.e. $Q_{ij}(s) \neq 0$ for $i \neq j$ and $P_{kl}(s) \neq 0$, $\forall(k, l)$, corresponds to a subsystem $S_m(s)$ in the structured linear fractional transformation. Each link can be realized in the same manner that the corresponding subsystem is realized and no pole-zero cancellations can occur in either since the states are all observable. Therefore, $n_{dsf} = n_{sub}$.

2. **Case 1: Shared Hidden States** Consider a system with h hidden states, where the i^{th} hidden state is shared in r_i links in $Q(s)$ and/or $P(s)$, where $r_1 > 1$, $\forall i$. Then, any minimal realization of the structured linear fractional transformation will have $n_{ns} + \sum_{i=1}^{h} r_i$ states, where n_{ns} are the non-shared hidden states in the system, while any minimal realization of the dynamical structure function will have $n_{ns} + \sum_{i=1}^{h} 1$ states. Since $r_1 > 1$, $\forall i$, that means $n_{dsf} < n_{sub}$

Therefore, for any arbitrary system $n_{dsf} \leq n_{sub}$. □

Theorem 9. *Consider a dynamical structure function* $(Q(s), P(s))$*, with* $G(s) = (I - Q(s))^{-1}P(s)$ *and* $(N, S(s))$ *the structured linear fractional transformation derived from the graphical dual of* $(Q(s), P(s))$*. Then the following relationship is true for the order of each minimal realization:*

$$\textit{McMillan degree} \leq \textit{structural degree} \leq \textit{subsystem degree} \tag{5.9}$$

Proof. From Lemma 9 we know that the McMillan degree is equal to the number of states that are both controllable and observable, n_{co}. Equation 5.6 then showed that McMillan degree $\leq$ structural degree. Finally, Lemma 10 shows that structural degree $\leq$ subsystem degree, which completes the proof. □

5.2 Network Semantics

Semantics is a branch of linguistics and logic concerned with *meaning* [5]. Remembering that a single instance of a system representation with low structural informativity will be consistent with a *set* of instances of a system representation with higher structural informativity, we call this set the *semantics* of the instance of the lower structural informativity representation.

Here we define the semantics of an instance of a system representation to be the set of state space realizations consistent with that instance. Each instance of a transfer

function, dynamical structure function, and structured linear fractional transformation has an associated set of state space realizations with consistent dynamics and structure.

The methodology we developed for comparing the semantics of the transfer function, structured linear fractional transformation, and dynamical structure function is as follows:

1. Given the dynamical structure function $(\hat{Q}(s), \hat{P}(s))$ of the form in (2.28) where $p = p_1$, meaning C is full row rank, and $D = 0$, calculate the associated transfer function $G(s)$.
2. Determine the structured linear fractional transformation $(N, S(s))$ that is the graphical dual of the dynamical structure function $(\hat{Q}(s), \hat{P}(s))$.
3. Determine the set of realizations associated with the transfer function, dynamical structure function, and structured linear fractional transformation.
4. Pad the set of realizations associated with the transfer function, $G(s)$, and the dynamical structure function, $(\hat{Q}(s), \hat{P}(s))$, with additional states to ensure all sets are the same order and, thus, comparable.
5. Compare the semantics, i.e. the resulting sets of realizations, of each system representation.

To start, assume we are given an instance of a dynamical structure function $(\hat{Q}(s), \hat{P}(s))$.

5.2.1 Step 1: Calculate the Transfer Function

Consider a dynamical structure function of the form (2.27) where $p = p_1$, meaning C is full row rank, and $D = 0$, meaning $(\hat{Q}(s), \hat{P}(s)) = (Q(s), P(s))$. The associated transfer function is

$$G(s) = (I - Q(s))^{-1} P(s)$$

where $(I - Q(s))$ is invertible because $Q(s)$ is a strictly proper, hollow matrix.

5.2.2 Step 2: Structured Linear Fractional Transformation as a Graphical Dual

Consider a state space model of the form:

$$\begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} A_1 & k_{12}B_1C_2 & ... & k_{1n}B_1C_n \\ k_{21}B_2C_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & k_{(n-1)n}B_{n-1}C_n \\ k_{n1}B_nC_1 & ... & k_{n(n-1)}B_nC_{n-1} & A_n \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \tag{5.10}$$

$$+ \begin{bmatrix} l_{11}B_1 & 0 & ... & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & ... & 0 & l_{nn}B_n \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} C_1 & 0 & ... & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & ... & 0 & C_n \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

where $k_{ij} \in \{0, 1\}$ is non-zero if there is a connection from the j^{th} to i^{th} subsystem, $l_{ii} \in \{0, 1\}$ is non-zero if the i^{th} subsystem is affected by an external input, and for each $i = 1, ..., q$, we have $A_i \in \mathbb{R}^{n_i \times n_i}$, $B_i \in \mathbb{R}^{n_i \times 1}$, and $C_i \in \mathbb{R}^{1 \times n_i}$. We assume that all transfer functions $G_i(s) = C_i(sI - A_i)^{-1}B_i$ are single-input single-output (SISO) which is reasonable for a comparison to the dynamical structure function as a graphical dual, also the state space representation in (5.10) makes two other assumptions to make it comparable to the dynamical structure function:

1. K is hollow, meaning that outputs of subsystems don't directly affect themselves and
2. L is diagonal, meaning each input targets a single subsystem.

These assumptions are a reasonable starting point for the development of the semantics of the structured linear fractional transformation and extending the results to structured linear fractional transformations with less restrictions will be the product of future work.

Given $(Q(s), P(s))$ of a system, the graphical dual can be created mathematically taking each non-zero entry of $Q(s)$ and $P(s)$ and setting it as a SISO subsystem $G_k(s)$ in $S(s)$. Utilizing the equations

$$N = \begin{bmatrix} 0 & I \\ L & K \end{bmatrix}, \qquad S(s) = \begin{bmatrix} G_1(s) & 0 & ... & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & ... & 0 & G_n(s) \end{bmatrix} \tag{5.11}$$

where $G_i(s) = C_i(sI - A_i)^{-1}B_i$,

$$L = \begin{bmatrix} l_{11} & 0 & .. & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & ... & 0 & l_{nn} \end{bmatrix}, \qquad K = \begin{bmatrix} 0 & k_{12} & ... & k_{1n} \\ k_{21} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & k_{(n-1)n} \\ k_{n1} & ... & k_{n(n-1)} & 0 \end{bmatrix} \tag{5.12}$$

given the values of l_{ii} and k_{ij} from (5.10). Then the structure of $\begin{bmatrix} L & K \end{bmatrix}$ is determined by setting $L_{ij} = 1$ if, given (2.28), the input $U_j(s)$ affects the measured state $Y_i(s)$, i.e. $P_{ij}(s) \neq 0$, and $K_{ij} = 1$ if $Y_j(s)$ affects $Y_i(s)$, i.e. $Q_{ij}(s) \neq 0$.

5.2.3 Step 3: Determine the Set of Realizations Associated with Each System Representation

First, we describe the set of realizations associated with the transfer function, a well known result in the systems theory literature [30]. This, in turn, will then inform the network

semantics of both the structured linear fractional transformation and the dynamical structure function.

Network Semantics of a Transfer Function

The semantics of the transfer function, i.e. the set of all realizations associated with the transfer function, is characterized by any invertible matrix $T \in \mathbb{R}^{n \times n}$, where n is the minimal order of the transfer function and T represents any change of basis of the state variables $x(t)$. Given a state space model that is a realization of the transfer function G of the form:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{5.13}$$

The associated transfer function is

$$G(s) = C(sI - A)^{-1}B + D$$

Applying the transformation T to the state space system in (5.13) we get:

$$\begin{aligned} \dot{z}(t) &= TAT^{-1}z(t) + TBu(t) \\ y(t) &= CT^{-1}z(t) + Du(t) \end{aligned} \tag{5.14}$$

The associated transfer function is given by:

$$\begin{aligned} \hat{G}(s) &= CT^{-1}(sI - TAT^{-1})^{-1}TB + D \\ &= CT^{-1}(sTT^{-1} - TAT^{-1})^{-1}TB + D \\ &= CT^{-1}(T(sI - A)T^{-1})^{-1}TB + D \\ &= CT^{-1}T(sI - A)^{-1}T^{-1}TB + D \\ &= C(sI - A)^{-1}B + D \\ &= G(s) \end{aligned}$$

Then the set of all realizations specifying the semantics of $G(s)$ is given by (5.14) for any invertible matrix T.

Network Semantics of a Structured Linear Fractional Transformation

In this section, we detail the set of transformations that maintain the class of structured linear fractional transformations defined in (5.11).

Lemma 11. *Given a state space model of the form (5.10), an invertible transformation T maintains the structured linear fractional transformation* $\Longleftrightarrow$

$$T = \begin{bmatrix} T_{11} & \dots & T_{1n} \\ \vdots & \ddots & \vdots \\ T_{n1} & \dots & T_{nn} \end{bmatrix}, \qquad T^{-1} = \begin{bmatrix} \hat{T}_{11} & \dots & \hat{T}_{1n} \\ \vdots & \ddots & \vdots \\ \hat{T}_{n1} & \dots & \hat{T}_{nn} \end{bmatrix} \tag{5.15}$$

where

1. *(Property A) T_{ii} is any invertible matrix for all i and $\hat{T}_{ii} = T_{ii}^{-1}$,*
2. *(Property B) $T_{ij}B_j = 0$ and $C_i\hat{T}_{ij} = 0$ for all i, j when $i \neq j$,*
3. *(Property C) For each i, j, h where $i \neq j$ or $i \neq h$ or $j \neq h$, we have $\sum_{j=1}^{n} T_{ij}A_j\hat{T}_{jh} = 0$*

Proof. First, we show necessity by taking a transformation T of the form (5.15) and showing that each of the Properties A, B, and C must be met to ensure the structured linear fractional transformation is maintained. Consider a transformation T, where T is any invertible matrix,

on a state space model of the form given in (5.10), which is:

$$\begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} A_1 & k_{12}B_1C_2 & \dots & k_{1n}B_1C_n \\ k_{21}B_2C_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & k_{(n-1)n}B_{n-1}C_n \\ k_{n1}B_nC_1 & \dots & k_{n(n-1)}B_nC_{n-1} & A_n \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$
$$+ \begin{bmatrix} l_{11}B_1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & l_{nn}B_n \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix}$$
$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} C_1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & C_n \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

Applying the transformation (5.15) to the B matrix, we get

$$\begin{aligned} TB &= \begin{bmatrix} T_{11} & \dots & T_{1n} \\ \vdots & \ddots & \vdots \\ T_{n1} & \dots & T_{nn} \end{bmatrix} \begin{bmatrix} l_{11}B_1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & l_{nn}B_n \end{bmatrix} \\ &= \begin{bmatrix} l_{11}T_{11}B_1 & \dots & l_{nn}T_{1n}B_n \\ \vdots & \ddots & \vdots \\ l_{11}T_{n1}B_1 & \dots & l_{nn}T_{nn}B_n \end{bmatrix} \end{aligned} \tag{5.16}$$

Since TB must remain a block diagonal matrix in order for the structured linear fractional transformation to be defined, $T_{ij}B_j = 0$ for all $i \neq j$.

Similarly, applying the transformation (5.15) to the C matrix, which gives

$$\begin{aligned} CT^{-1} &= \begin{bmatrix} C_1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & C_n \end{bmatrix} \begin{bmatrix} \hat{T}_{11} & \dots & \hat{T}_{1n} \\ \vdots & \ddots & \vdots \\ \hat{T}_{n1} & \dots & \hat{T}_{nn} \end{bmatrix} \\ &= \begin{bmatrix} C_1\hat{T}_{11} & \dots & C_1\hat{T}_{1n} \\ \vdots & \ddots & \vdots \\ C_n\hat{T}_{n1} & \dots & C_n\hat{T}_{nn} \end{bmatrix} \end{aligned} \tag{5.17}$$

since CT^{-1} must remain a block diagonal matrix in order for the structured linear fractional transformation to be defined, $C_i\hat{T}_{ij} = 0$ for all $i \neq j$. Moreover, taking the results from (5.16) and (5.17) we know that for each $\hat{T}_{ii} = T_{ii}^{-1}$ in order for the structured linear fractional transformation to be maintained, meaning each T_{ii} is invertible. Thus, we have shown that Properties A and B must be true in order for the structured linear fractional transformation to be maintained. The final step is to demonstrate that Property C must be true for the structured linear fractional transformation to be maintained.

Applying the transformation (5.15) to A yields:

$$\begin{aligned} TAT^{-1} &= \begin{bmatrix} T_{11} & \dots & T_{1n} \\ \vdots & \ddots & \vdots \\ T_{n1} & \dots & T_{nn} \end{bmatrix} \begin{bmatrix} A_1 & k_{12}B_1C_2 & \dots & k_{1n}B_1C_n \\ k_{21}B_2C_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & k_{(n-1)n}B_{n-1}C_n \\ k_{n1}B_nC_1 & \dots & k_{n(n-1)}B_nC_{n-1} & A_n \end{bmatrix} \begin{bmatrix} T_{11}^{-1} & \dots & \hat{T}_{1n} \\ \vdots & \ddots & \vdots \\ \hat{T}_{n1} & \dots & T_{nn}^{-1} \end{bmatrix} \\ &= \begin{bmatrix} T_{11}A_1 + \sum_{r=2}^{n} k_{r1}T_{1r}B_rC_1 & \dots & T_{1n}A_n + \sum_{r=1}^{n-1} k_{rn}T_{1r}B_rC_n \\ \vdots & \ddots & \vdots \\ T_{n1}A_1 + \sum_{r=2}^{n} k_{r1}T_{r1}B_rC_1 & \dots & T_{nn}A_n + \sum_{r=1}^{n-1} k_{rn}T_{r1}B_rC_n \end{bmatrix} \begin{bmatrix} T_{11}^{-1} & \dots & \hat{T}_{1n} \\ \vdots & \ddots & \vdots \\ \hat{T}_{n1} & \dots & T_{nn}^{-1} \end{bmatrix} \end{aligned} \tag{5.18}$$

Since $T_{ij}B_j = 0$ for all $i \neq j$, we have:

$$
\begin{aligned}
TAT^{-1} &= \begin{bmatrix} T_{11}A_1 & \dots & T_{1n}A_n + k_{1n}T_{11}B_1C_n \\ \vdots & \ddots & \vdots \\ T_{n1}A_1 + k_{n1}T_{nn}B_nC_1 & \dots & T_{nn}A_n \end{bmatrix} \begin{bmatrix} T_{11}^{-1} & \dots & \hat{T}_{1n} \\ \vdots & \ddots & \vdots \\ \hat{T}_{n1} & \dots & T_{nn}^{-1} \end{bmatrix} \\
&= \begin{bmatrix} T_{11}A_1T_{11}^{-1} + \sum_{r=2}^{n}(T_{1r}A_r\hat{T}_{r1} + k_{1r}T_{11}B_1C_r\hat{T}_{r1}) & \dots & T_{11}A_1\hat{T}_{1n} + \sum_{r=2}^{n}(T_{1r}A_r\hat{T}_{rn} + k_{1r}T_{11}B_1C_r\hat{T}_{rn}) \\ \vdots & \ddots & \vdots \\ T_{nn}A_n\hat{T}_{n1} + \sum_{r=1}^{n-1}(T_{nr}A_r\hat{T}_{r1} + k_{nr}T_{nn}B_nC_r\hat{T}_{r1}) & \dots & T_{nn}A_nT_{nn}^{-1} + \sum_{r=1}^{n-1}(T_{nr}A_r\hat{T}_{1r} + k_{nr}T_{nn}B_nC_r\hat{T}_{1r}) \end{bmatrix}
\end{aligned}
\tag{5.19}
$$

Since $C_i\hat{T}_{ji} = 0$ for all $i \neq j$, we have:

$$
TAT^{-1} = \begin{bmatrix} \sum_{r=1}^{n}(T_{1r}A_r\hat{T}_{r1}) & \dots & k_{1n}T_{11}B_1C_nT_{nn}^{-1} + \sum_{r=1}^{n}(T_{1r}A_r\hat{T}_{rn}) \\ \vdots & \ddots & \vdots \\ k_{n1}T_{nn}B_nC_1T_{11}^{-1} + \sum_{r=1}^{n}(T_{nr}A_r\hat{T}_{r1}) & \dots & \sum_{r=1}^{n}(T_{nr}A_r\hat{T}_{rn}) \end{bmatrix} \tag{5.20}
$$

Consider the transformed matrices TB and CT^{-1}. We know that for each i we have $\hat{B}_i = T_{ii}B_i$ and $\hat{C}_i = C_iT_{ii}^{-1}$, which means that in order to maintain the structured linear fractional transformation, the off diagonal entries of the transformed matrix TAT^{-1} must be $\hat{A}_{ij} = T_{ii}B_iC_jT_{jj}^{-1}$ and the diagonal entries must be $\hat{A}_{ii} = T_{ii}A_iT_{ii}^{-1}$. This means that for each i, j, h where $i \neq j$ or $i \neq h$ or $j \neq h$ we have $\sum_{j=1}^{n} T_{ij}A_j\hat{T}_{jh} = 0$, which means we have shown necessity of Property C.

Now, in order to show sufficiency we need to show that given a transformation (5.15) with Properties A, B, and C, that the structured linear fractional transformation is maintained. Applying a transformation (5.15) with these properties to a state space model of the form

(5.10) yields the state space model:

$$\begin{bmatrix} \dot{z}_1 \\ \vdots \\ \dot{z}_n \end{bmatrix} = \begin{bmatrix} T_{11}A_1T_{11}^{-1} & k_{12}T_{11}B_1C_2T_{22}^{-1} & \dots & k_{1n}T_{11}B_1C_nT_{nn}^{-1} \\ k_{21}T_{22}B_2C_1T_{11}^{-1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & k_{(n-1)n}T_{(n-1)(n-1)}B_{n-1}C_nT_{nn}^{-1} \\ k_{n1}T_{nn}B_nC_1T_{11}^{-1} & \dots & k_{n(n-1)}T_{nn}B_nC_{n-1}T_{(n-1)(n-1)}^{-1} & T_{nn}A_nT_{11}^{-1} \end{bmatrix} \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix}$$
$$+ \begin{bmatrix} l_{11}T_{11}B_1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & l_{nn}T_{nn}B_n \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix}$$
$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} C_1T_{11}^{-1} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & C_nT_{nn}^{-1} \end{bmatrix} \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} \tag{5.21}$$

The associated structured linear fractional transformation of (5.21) is given by:

$$\hat{S}(s) = \begin{bmatrix} \hat{G}_1(s) & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \hat{G}_q(s) \end{bmatrix} \tag{5.22}$$

where

$$\begin{aligned} \hat{G}_i(s) &= C_iT_{ii}^{-1}(sI - T_{ii}A_iT_{ii}^{-1})^{-1}T_{ii}B_i \\ &= C_i(sI - A_i)^{-1}B_i \\ &= G_i(s) \end{aligned} \tag{5.23}$$

which means $\hat{S}(s) = S(s)$ and

$$\hat{N} = \begin{bmatrix} 0 & I \\ \hat{L} & \hat{K} \end{bmatrix} \tag{5.24}$$

with $\hat{K} = K$ since the corresponding k values in the entries of the transformed A matrix do not change and $\hat{L} = L$ since the transformation does not affect the boolean structure of inputs that affect the subsystems, which means $\hat{N} = N$. Since $(\hat{S}(s), \hat{N}) = (S(s), N)$ the structured linear fractional transformation is maintained, which completes the proof. □

Theorem 10. *A transformation of the form*

$$T = \begin{bmatrix} T_{11} & 0 & ... & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & ... & 0 & T_{nn}^{-1} \end{bmatrix} \tag{5.25}$$

is necessary and sufficient for maintaining the structured linear fractional transformation.

Proof. The basic idea of the proof is that any state space model that can be reached by a transformation of the form

$$T = \begin{bmatrix} T_{11} & ... & T_{1n} \\ \vdots & \ddots & \vdots \\ T_{n1} & ... & T_{nn} \end{bmatrix}, \qquad T^{-1} = \begin{bmatrix} \hat{T}_{11} & ... & \hat{T}_{1n} \\ \vdots & \ddots & \vdots \\ \hat{T}_{n1} & ... & \hat{T}_{nn} \end{bmatrix} \tag{5.26}$$

with Properties A, B, and C from Lemma 11 can be reached using the transformation in (5.25). Applying the transformation (5.25) to a state space model of the form (5.10) yields:

$$\begin{bmatrix} \dot{z}_1 \\ \vdots \\ \dot{z}_n \end{bmatrix} = \begin{bmatrix} T_{11}A_1T_{11}^{-1} & k_{12}T_{11}B_1C_2T_{22}^{-1} & \dots & k_{1n}T_{11}B_1C_nT_{nn}^{-1} \\ k_{21}T_{22}B_2C_1T_{11}^{-1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & k_{(n-1)n}T_{(n-1)(n-1)}B_{n-1}C_nT_{nn}^{-1} \\ k_{n1}T_{nn}B_nC_1T_{11}^{-1} & \dots & k_{n(n-1)}T_{nn}B_nC_{n-1}T_{(n-1)(n-1)}^{-1} & T_{nn}A_nT_{11}^{-1} \end{bmatrix} \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} + \begin{bmatrix} l_{11}T_{11}B_1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & l_{nn}T_{nn}B_n \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} C_1T_{11}^{-1} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & C_nT_{nn}^{-1} \end{bmatrix} \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} \tag{5.27}$$

which, by Lemma 11, maintains the structured linear fractional transformation. Since (5.27) is equivalent to the system in (5.21), which can be reached by a transformation of the form (5.26), the proof is complete. □

Network Semantics of a Dynamical Structure Function

When calculating the dynamical structure function from a state space model, there exists an intermediate representation for $W(s)$ and $V(s)$, as defined in (2.23) and (2.24). In this section, we detail sufficient conditions for a transformation that maintains $W(s)$ and $V(s)$ for the case when C is full row rank and $D = 0$, which is in turn a sufficient condition for a transformation that maintains a dynamical structure function. Necessary conditions for a transformation that maintains a dynamical structure function is the subject of future work.

Maintaining $W(s)$ and $V(s)$ when C is Full Row Rank Consider a state space model of the form (2.20) with $p = p_1$, which means C full row rank, and $D = 0$, which we denote

$(A, B, C, 0)$. This uniquely defines a pair of the intermediate representation of the dynamical structure function $(W(s), V(s))$ through a transformation T as defined in (2.18). Next, consider a transformation $\hat{T}$ on (2.20) that yields a new state space model $(\hat{A}, \hat{B}, \hat{C}, 0)$ which in turn uniquely defines a new pair of the intermediate representation of the dynamical structure function $(\bar{W}(s), \bar{V}(s))$ through a transformation $\bar{T}$. We want to show the properties of $\hat{T}$ such that

$$(W(s), V(s)) = (\bar{W}(s), \bar{V}(s))$$

which is a sufficient condition for

$$(Q(s), P(s)) = (\bar{Q}(s), \bar{P}(s))$$

The various relationships between state space realizations given by the transformations T, $\hat{T}$ and $\bar{T}$ can be summarized as in Figure 5.3.

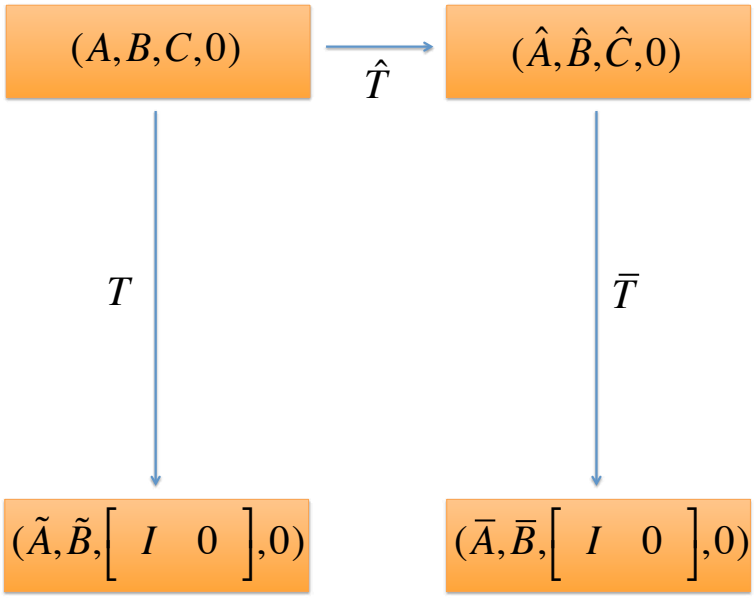


Figure 5.3: The relationship between the state space representations in Lemma 12.

1. **Determining Formulas for** $(W(s), V(s))$

Consider the state space system

$$(A, B, C, 0) \tag{5.28}$$

where C is full row rank. The transformation

$$T = \begin{bmatrix} C \\ E^T \end{bmatrix} \tag{5.29}$$

on the system, where E is any basis of the null space of C, yields the system

$$(\tilde{A}, \tilde{B}, \tilde{C}, 0) = (TAT^{-1}, TB, CT^{-1}, 0) \tag{5.30}$$

where

$$T^{-1} = \begin{bmatrix} C^T(CC^T)^{-1} & E \end{bmatrix}. \tag{5.31}$$

Applying the transformation (5.29) to (5.28) yields the following equations:

$$\begin{aligned} \tilde{A} = TAT^{-1} &= \begin{bmatrix} C \\ E^T \end{bmatrix} A \begin{bmatrix} C^T(CC^T)^{-1} & E \end{bmatrix} \\ &= \begin{bmatrix} CAC^T(CC^T)^{-1} & CAE \\ E^TAC^T(CC^T)^{-1} & E^TAE \end{bmatrix} \end{aligned} \tag{5.32}$$

$$\begin{aligned} \tilde{B} = TB &= \begin{bmatrix} C \\ E^T \end{bmatrix} B \\ &= \begin{bmatrix} CB \\ E^TB \end{bmatrix} \end{aligned} \tag{5.33}$$

$$\tilde{C} = CT^{-1} = C\begin{bmatrix} C^T(CC^T)^{-1} & E \end{bmatrix}$$

$$= \begin{bmatrix} CC^T(CC^T)^{-1} & CE \end{bmatrix} \tag{5.34}$$

$$= \begin{bmatrix} I & 0 \end{bmatrix}$$

From (5.32), (5.33), (5.34), and the definition of the intermediate representation of the dynamical structure function in (2.23) and (2.24) we get

$$W(s) = \tilde{A}_{11} + \tilde{A}_{12}(sI - \tilde{A}_{22})^{-1}\tilde{A}_{21}$$

$$= CAC^T(CC^T)^{-1} + CAE(sI - E^TAE)^{-1}E^TAC^T(CC^T)^{-1} \tag{5.35}$$

$$= CAC^T(CC^T)^{-1} + CAE(E^T(sI - A)E)^{-1}E^TAC^T(CC^T)^{-1}$$

$$V(s) = \tilde{B}_1 + \tilde{A}_{12}(sI - \tilde{A}_{22})^{-1}\tilde{B}_2$$

$$= CB + CAE(sI - E^TAE)^{-1}E^TB \tag{5.36}$$

$$= CB + CAE(E^T(sI - A)E)^{-1}E^TB$$

2. **Deriving Formulas for** $(\bar{W}(s), \bar{V}(s))$

Consider an invertible matrix $\hat{T}$ that transforms the system in (5.28) such that we get

$$(\hat{A}, \hat{B}, \hat{C}, 0) = (\hat{T}A\hat{T}^{-1}, \hat{T}B, C\hat{T}^{-1}, 0). \tag{5.37}$$

In order to determine the corresponding dynamical structure function, we must apply a state transformation of the form

$$\bar{T} = \begin{bmatrix} \hat{C} \\ \hat{E}^T \end{bmatrix} = \begin{bmatrix} C\hat{T}^{-1} \\ \hat{E}^T \end{bmatrix} \tag{5.38}$$

where $\hat{E}$ is any basis of the null space of $C\hat{T}^{-1}$. Note that

$$\bar{T}^{-1} = \begin{bmatrix} \hat{T}^{-T}C^T(C\hat{T}^{-1}\hat{T}^{-T}C^T)^{-1} & \hat{E} \end{bmatrix}. \tag{5.39}$$

Applying the transformation (5.38) to the state space model (5.37), we get the following equations:

$$\begin{aligned} \bar{A} = \bar{T}\hat{A}\bar{T}^{-1} = \bar{T}\hat{T}A\hat{T}^{-1}\bar{T}^{-1} &= \begin{bmatrix} C\hat{T}^{-1} \\ \hat{E}^T \end{bmatrix} \hat{T}A\hat{T}^{-1} \begin{bmatrix} \hat{T}^{-T}C^T(C\hat{T}^{-1}\hat{T}^{-T}C^T)^{-1} & \hat{E} \end{bmatrix} \\ &= \begin{bmatrix} CA\hat{T}^{-1}\hat{T}^{-T}C^T(C\hat{T}^{-1}\hat{T}^{-T}C^T)^{-1} & CA\hat{T}^{-1}\hat{E} \\ \hat{E}^T\hat{T}A\hat{T}^{-1}\hat{T}^{-T}C^T(C\hat{T}^{-1}\hat{T}^{-T}C^T)^{-1} & \hat{E}^T\hat{T}A\hat{T}^{-1}\hat{E} \end{bmatrix} \end{aligned} \tag{5.40}$$

$$\begin{aligned} \bar{B} = \bar{T}\hat{B} = \bar{T}\hat{T}B &= \begin{bmatrix} C\hat{T}^{-1} \\ \hat{E}^T \end{bmatrix} \hat{T}B \\ &= \begin{bmatrix} CB \\ \hat{E}^T\hat{T}B \end{bmatrix} \end{aligned} \tag{5.41}$$

$$\bar{C} = \hat{C}\bar{T}^{-1} = C\hat{T}^{-1}\bar{T}^{-1} = C\hat{T}^{-1}\begin{bmatrix}\hat{T}^{-T}C^T(C\hat{T}^{-1}\hat{T}^{-T}C^T)^{-1} & \hat{E}\end{bmatrix}$$

$$= \begin{bmatrix}C\hat{T}^{-1}\hat{T}^{-T}C^T(C\hat{T}^{-1}\hat{T}^{-T}C^T)^{-1} & C\hat{T}^{-1}\hat{E}\end{bmatrix} \tag{5.42}$$

$$= \begin{bmatrix}I & 0\end{bmatrix}$$

From (5.40), (5.41), (5.42) and the definition of the intermediate representation of the dynamical structure function in (2.23) and (2.24), we get

$$\begin{aligned}\bar{W}(s) &= \bar{A}_{11} + \bar{A}_{12}(sI - \bar{A}_{22})^{-1}\bar{A}_{21} \\ &= CA\hat{T}^{-1}\hat{T}^{-T}C^T(C\hat{T}^{-1}\hat{T}^{-T}C^T)^{-1} + \\ &\quad CA\hat{T}^{-1}\hat{E}(sI - \hat{E}^T\hat{T}A\hat{T}^{-1}\hat{E})^{-1}\hat{E}^T\hat{T}A\hat{T}^{-1}\hat{T}^{-T}C^T(C\hat{T}^{-1}\hat{T}^{-T}C^T)^{-1} \\ &= CA\hat{T}^{-1}\hat{T}^{-T}C^T(C\hat{T}^{-1}\hat{T}^{-T}C^T)^{-1} + \\ &\quad CA\hat{T}^{-1}\hat{E}(\hat{E}^T\hat{T}(sI - A)\hat{T}^{-1}\hat{E})^{-1}\hat{E}^T\hat{T}A\hat{T}^{-1}\hat{T}^{-T}C^T(C\hat{T}^{-1}\hat{T}^{-T}C^T)^{-1}\end{aligned} \tag{5.43}$$

$$\begin{aligned}\bar{V}(s) &= \bar{B}_1 + \bar{A}_{12}(sI - \bar{A}_{22})^{-1}\bar{B}_2 \\ &= CB + CA\hat{T}^{-1}E(sI - \hat{E}^T\hat{T}A\hat{T}^{-1}\hat{E})^{-1}\hat{E}^T\hat{T}B \\ &= CB + CA\hat{T}^{-1}E(\hat{E}^T\hat{T}(sI - A)\hat{T}^{-1}\hat{E})^{-1}\hat{E}^T\hat{T}B\end{aligned} \tag{5.44}$$

Given formulas for $(W(s), V(s))$ and $(\bar{W}(s), \bar{V}(s))$, Lemma 12 outlines necessary and sufficient conditions for maintaining this intermediate representation.

Lemma 12. $\hat{T}^{-1}\hat{T}^{-T} = I + \bar{E} \implies (W(s), V(s)) = (\bar{W}(s), \bar{V}(s))$, *where* $\bar{E} = \bar{E}^T$ *and* $range(\bar{E}) \subseteq \mathscr{N}(C)$.

Proof. Note that we assume $\hat{T}^{-1}\hat{T}^{-T} = I + \bar{E}$, where $\bar{E} = \bar{E}^T$. This is because $\hat{T}^{-1}\hat{T}^{-T}$ is symmetric, which means $I + \bar{E}$ is symmetric $\implies \bar{E} = \bar{E}^T$.

The proof will proceed as follows:

1. Detail the equations $(W(s), V(s)) = (\bar{W}(s), \bar{V}(s))$.
2. Split the two equations $W(s) = \bar{W}(s)$ and $V(s) = \bar{V}(s)$ into four equations, two equations each, based on the dynamic and static parts of each.
3. Analyze the corresponding equations to show sufficient conditions on $\hat{T}$ that maintain the dynamical structure function.

First, the comparison of $(W(s), V(s))$ and $(\bar{W}(s), \bar{V}(s))$ yields

$$\begin{gathered} W(s) = \bar{W}(s) \implies \\ CAC^T(CC^T)^{-1} + CAE(E^T(sI - A)E)^{-1}E^TAC^T(CC^T)^{-1} = \\ CA\hat{T}^{-1}\hat{T}^{-T}C^T(C\hat{T}^{-1}\hat{T}^{-T}C^T)^{-1} + \\ CA\hat{T}^{-1}\hat{E}(\hat{E}^T\hat{T}(sI - A)\hat{T}^{-1}\hat{E})^{-1}\hat{E}^T\hat{T}A\hat{T}^{-1}\hat{T}^{-T}C^T(C\hat{T}^{-1}\hat{T}^{-T}C^T)^{-1} \end{gathered} \tag{5.45}$$

$$\begin{gathered} V(s) = \bar{V}(s) \implies \\ CB + CAE(E^T(sI - A)E)^{-1}E^TB = \\ CB + CA\hat{T}^{-1}E(\hat{E}^T\hat{T}(sI - A)\hat{T}^{-1}\hat{E})^{-1}\hat{E}^T\hat{T}B \end{gathered} \tag{5.46}$$

Since each of $W(s), V(s), \bar{W}(s)$ and $\bar{V}(s)$ are transfer functions, we can split the equations (5.45) and (5.46) into their dynamic and static parts to get:

$$CAC^T(CC^T)^{-1} = CA\hat{T}^{-1}\hat{T}^{-T}C^T(C\hat{T}^{-1}\hat{T}^{-T}C^T)^{-1} \tag{5.47}$$

$$CAE(E^T(sI-A)E)^{-1}E^TAC^T(CC^T)^{-1} = \\ CA\hat{T}^{-1}\hat{E}(\hat{E}^T\hat{T}(sI-A)\hat{T}^{-1}\hat{E})^{-1}\hat{E}^T\hat{T}A\hat{T}^{-1}\hat{T}^{-T}C^T(C\hat{T}^{-1}\hat{T}^{-T}C^T)^{-1} \tag{5.48}$$

$$CB = CB \tag{5.49}$$

$$CAE(E^T(sI-A)E)^{-1}E^TB = CA\hat{T}^{-1}E(\hat{E}^T\hat{T}(sI-A)\hat{T}^{-1}\hat{E})^{-1}\hat{E}^T\hat{T}B \tag{5.50}$$

Ignoring equation (5.49), since it does not contain $\hat{T}$ and doesn't convey any useful relationship, we focus instead on equations (5.47), (5.48), and (5.50). We begin by analyzing equation (5.47). Define $R = \hat{T}^{-1}\hat{T}^{-T}$ to get the equation

$$CAC^T(CC^T)^{-1} = CARC^T(CRC^T)^{-1} \tag{5.51}$$

Subtracting $CAC^T(CC^T)^{-1}$ from both sides yields

$$CARC^T(CRC^T)^{-1} - CAC^T(CC^T)^{-1} = 0 \tag{5.52}$$

Then factoring out CA on the left gives us

$$CA\left[RC^T(CRC^T)^{-1} - C^T(CC^T)^{-1}\right] = 0 \tag{5.53}$$

which means

$$RC^T(CRC^T)^{-1} - C^T(CC^T)^{-1} = \tilde{E}, \tag{5.54}$$

where $range(\tilde{E}) \subseteq \mathcal{N}(CA)$. Equation 5.54 can be rewritten as

$$RC^T(CRC^T)^{-1} = C^T(CC^T)^{-1} + \tilde{E}. \tag{5.55}$$

Multiplying (5.55) on the left by C gives us

$$CRC^T(CRC^T)^{-1} = CC^T(CC^T)^{-1} + C\tilde{E} \tag{5.56}$$

$$I = I + C\tilde{E} \tag{5.57}$$

$$C\tilde{E} = 0 \tag{5.58}$$

which means $range(\tilde{E}) \subseteq \mathscr{N}(C)$, a fact that will be necessary for defining conditions on $\hat{T}$ that allow it to maintain the intermediate representation of the dynamical structure function.

The final step to determine conditions on $\hat{T}$ comes from (5.55) and letting

$$R = G + H. \tag{5.59}$$

Determining the values of G and H will give us the conditions on $\hat{T}$ that maintain the dynamical structure functions. First, we get the equation

$$(G+H)C^T(C(G+H)C^T)^{-1} = C^T(CC^T)^{-1} + \tilde{E} \tag{5.60}$$

$$GC^T(C(G+H)C^T)^{-1} + HC^T(C(G+H)C^T)^{-1} = C^T(CC^T)^{-1} + \tilde{E} \tag{5.61}$$

While there are potentially infinite values for G and H, we only care what their sum is in order to have a value for R, which in turn details conditions for $\hat{T}$. Therefore, we only need to know one possible value for G and H in order to determine their sum, which gives us R. One possible value for G and H comes by splitting (5.61) into the following two equations:

$$GC^T(C(G+H)C^T)^{-1} = C^T(CC^T)^{-1} \tag{5.62}$$

$$HC^T(C(G+H)C^T)^{-1} = \tilde{E} \tag{5.63}$$

First, we get a value for H from (5.63) by multiplying on the left by C, which gives us:

$$CHC^T(C(G+H)C^T)^{-1} = 0 \tag{5.64}$$

$$CHC^T = 0 \tag{5.65}$$

Thus a **sufficient** condition is that

$$range(H) \subseteq \mathcal{N}(C). \tag{5.66}$$

Applying the result from (5.66) to (5.62) yields:

$$GC^T(CGC^T)^{-1} = C^T(CC^T)^{-1} \tag{5.67}$$

$$GC^T(CGC^T)^{-1}CC^T = C^T \tag{5.68}$$

$$GC^T(CGC^T)^{-1}CC^T - C^T = 0 \tag{5.69}$$

$$(G-I)C^T\left[(CGC^T)^{-1}CC^T - I\right] = 0 \tag{5.70}$$

which means either

$$range(G-I) \subseteq \mathcal{N}(C) \tag{5.71}$$

or

$$range((CGC^T)^{-1}CC^T - I) \subseteq \mathcal{N}(C^T). \tag{5.72}$$

Exploring (5.72) further yields:

$$C^T(CGC^T)^{-1}CC^T - C^T = 0 \tag{5.73}$$

Combining (5.69) and (5.73) we get

$$GC^T(CGC^T)^{-1}CC^T = C^T(CGC^T)^{-1}CC^T \tag{5.74}$$

$$GC^T = C^T \tag{5.75}$$

$$(G - I)C^T = 0 \tag{5.76}$$

Therefore, the condition from (5.72) is equivalent to the condition from (5.71), i.e.

$$range(G - I) \subseteq \mathcal{N}(C)$$

This means

$$G = I + L \tag{5.77}$$

where $range(L) \subseteq \mathcal{N}(C)$. Combining (5.59), (5.66) and (5.77) and remembering that we defined $R = \hat{T}^{-1}\hat{T}^{-T}$ we get

$$\begin{aligned} \hat{T}^{-1}\hat{T}^{-T} &= R \\ &= G + H \\ &= I + L + H \\ &= I + \bar{E} \end{aligned} \tag{5.78}$$

where $\bar{E} = L + H \implies range(\bar{E}) \subseteq \mathcal{N}(C)$.

Although this is the result we want, we are not done yet, we need to ensure that this result is consistent with the equations in (5.48) and (5.50). In order to do so, we must first show that $\hat{T}^{-1}\hat{E}$ and $\hat{T}^{T}\hat{E}$ are bases of the null space of C. Consider the transformation $\bar{T}\hat{T}$, this is a transformation that maps C to $\begin{bmatrix} I & 0 \end{bmatrix}$.

$$\begin{aligned} \bar{T}\hat{T} &= \begin{bmatrix} \hat{C} \\ \hat{E}^{T} \end{bmatrix} \hat{T} \\ &= \begin{bmatrix} C\hat{T}^{-1} \\ \hat{E}^{T} \end{bmatrix} \hat{T} \\ &= \begin{bmatrix} C \\ \hat{E}^{T}\hat{T} \end{bmatrix} \end{aligned} \tag{5.79}$$

From (2.18) we know that $(\hat{E}^{T}\hat{T})^{T}$ is a basis for the null space of C. We will define

$$\breve{E} = (\hat{E}^{T}\hat{T})^{T} \tag{5.80}$$

Now, we want to show that $\hat{T}^{-1}\hat{E}$ is a basis of the null space of C and that $\breve{E} = \hat{T}^{-1}\hat{E}$, i.e. $(\hat{E}^{T}\hat{T})^{T} = \hat{T}^{-1}\hat{E}$. We start by looking at the inverse of $\bar{T}\hat{T}$, which is:

$$\begin{aligned} \hat{T}^{-1}\bar{T}^{-1} &= \hat{T}^{-1}\begin{bmatrix} \hat{T}^{-T}C^{T}(C\hat{T}^{-1}\hat{T}^{-T}C^{T}) & \hat{E} \end{bmatrix} \\ &= \begin{bmatrix} \hat{T}^{-1}\hat{T}^{-T}C^{T}(C\hat{T}^{-1}\hat{T}^{-T}C^{T}) & \hat{T}^{-1}\hat{E} \end{bmatrix} \end{aligned} \tag{5.81}$$

Since we know $\hat{T}^{-1}\hat{T}^{-T} = I + \bar{E}$, we have

$$\hat{T}^{-1}\bar{T}^{-1} = \begin{bmatrix} C^{T}(CC^{T}) & \hat{T}^{-1}\hat{E} \end{bmatrix} \tag{5.82}$$

which means that $\hat{T}^{-1}\hat{E}$ is a basis of the null space of C and that $\breve{E} = \hat{T}^{-1}\hat{E}$. Given this, (5.48) and (5.50) can be rewritten as

$$\begin{aligned} CAE(E^T(sI-A)E)^{-1}E^TAC^T(CC^T)^{-1} = \\ CA\breve{E}(\breve{E}^T(sI-A)\breve{E})^{-1}\breve{E}^TAC^T(CC^T)^{-1} \end{aligned} \tag{5.83}$$

$$CAE(E^T(sI-A)E)^{-1}E^TB = CA\breve{E}(\breve{E}^T(sI-A)\breve{E})^{-1}\breve{E}^TB \tag{5.84}$$

Theorem 2 demonstrated that $W(s)$ and $V(s)$ are invariant to basis of the null space of C, meaning (5.83) and (5.84) are consistent, which completes the proof.

□

Theorem 11. *Consider a system $(A, B, C, 0)$ with C full row rank whose dynamical structure function is given by $(Q(s), P(s))$. Then the system $(\hat{A}, \hat{B}, \hat{C}, 0) = (\hat{T}A\hat{T}^{-1}, \hat{T}B, C\hat{T}^{-1}, 0)$ has an associated dynamical structure function $(\bar{Q}(s), \bar{P}(s))$. Then*

$$\hat{T}^{-1}\hat{T}^{-T} = I + \bar{E} \implies (Q(s), P(s)) = (\bar{Q}(s), \bar{P}(s))$$

where $\bar{E} = \bar{E}^T$ and $range(\bar{E}) \subseteq \mathscr{N}(C)$.

Proof. If $\hat{T}^{-1}\hat{T}^{-T} = I + \bar{E}$, where $\bar{E} = \bar{E}^T$ $range(\bar{E}) \subseteq \mathscr{N}(C)$, then $(W(s), V(s)) = (\bar{W}(s), \bar{V}(s))$ as shown in Lemma 12. By the definition of the dynamical structure function, $W(s) = \bar{W}(s) \implies Q(s) = \bar{Q}(s)$. □

Corollary 2. *If $\hat{T}$ is orthogonal, then $(Q(s), P(s)) = (\bar{Q}(s), \bar{P}(s))$.*

Proof. From Theorem 11, we know that $\hat{T}^{-1}\hat{T}^{-T} = I + \bar{E} \implies (Q(s), P(s)) = (\bar{Q}(s), \bar{P}(s))$, where $range(\bar{E}) \subseteq \mathscr{N}(C)$ and $\bar{E} = \bar{E}^T$. Since $\bar{E} = 0$ meets these conditions, we have $\hat{T}^{-1}\hat{T}^{-T} = I \implies (Q(s), P(s)) = (\bar{Q}(s), \bar{P}(s))$. By definition, $\hat{T}^{-1}\hat{T}^{-T} = I$ means that $\hat{T}$ is orthogonal, which completes the proof. □

Example 11. *Consider the following state space system:*

$$\begin{aligned} \dot{x} &= \begin{bmatrix} -1 & 0 & 1 \\ 1 & -2 & 0 \\ 0 & 1 & -3 \end{bmatrix} x + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \end{bmatrix} x \end{aligned} \tag{5.85}$$

From the definition of the extended dynamical structure function, (2.18) states that

$$T = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 1 & -2 & 1 \end{bmatrix} \tag{5.86}$$

must be applied this transformation to (5.85) to get the state space model:

$$\begin{aligned} \dot{z} &= \begin{bmatrix} -10.83 & 6.333 & -0.8333 \\ -12.83 & 7.333 & -0.8333 \\ 1.5 & 1 & -2.5 \end{bmatrix} z + \begin{bmatrix} 1 & 4 & 9 \\ 2 & 6 & 12 \\ 1 & -4 & 3 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} z \end{aligned} \tag{5.87}$$

with corresponding dynamical structure function

$$Q(s) = \begin{bmatrix} 0 & \frac{6.333s^2+32.5s+41.67}{s^3+15.83s^2+61.67s+70.83} \\ \frac{-12.83s^2-65.42s-83.33}{s^3-2.333s^2-31.25s-47.92} & 0 \end{bmatrix}$$

$$P(s) = \begin{bmatrix} \frac{s^2+4.167s+4.167}{s^3+15.83s^2+61.67s+70.83} & \frac{4s^2+23.33s+33.33}{s^3+15.83s^2+61.67s+70.83} & \frac{9s^2+42.5s+50}{s^3+15.83s^2+61.67s+70.83} \\ \frac{2s^2+9.167s+10.42}{s^3-2.333s^2-31.25s-47.92} & \frac{6s^2+33.33s+45.83}{s^3-2.333s^2-31.25s-47.92} & \frac{12s^2+57.5s+68.75}{s^3-2.333s^2-31.25s-47.92} \end{bmatrix} \quad (5.88)$$

Now, one possible transformation that maintains the dynamical structure function when applied to the original system in (5.85) can be created as follows:

1. *Find a vector in the nullspace of C, in this case* $\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$
2. *Create a symmetric matrix $\bar{E}$ using this vector, such as* $\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$
3. *Add I to the matrix $\bar{E}$ to get* $\hat{T}^{-1}\hat{T}^{-T} = R = I + \bar{E} = \begin{bmatrix} 2 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 2 \end{bmatrix}$
4. *One way to get $\hat{T}$ is to set* $\hat{T}^{-1} = \sqrt{R} = \begin{bmatrix} 1.2743 & -0.5486 & 0.2743 \\ -0.5486 & 2.0972 & -0.5486 \\ 0.2743 & -0.5486 & 1.2743 \end{bmatrix}$
5. *Then* $\hat{T} = \begin{bmatrix} 0.8963 & 0.2073 & -0.1037 \\ 0.2073 & 0.5853 & 0.2073 \\ -0.1037 & 0.2073 & 0.8963 \end{bmatrix}$

Note that $\hat{T}$ is symmetric, but not orthogonal.

Applying this transformation to the original system in (5.85) yields

$$\begin{aligned} \dot{x} &= \begin{bmatrix} -0.2624 & -1.371 & 1.634 \\ 0.8963 & -2 & 0.1037 \\ -0.6339 & 2.371 & -3.738 \end{bmatrix} x + \begin{bmatrix} 0.8963 & 0.4147 & -0.311 \\ 0.2073 & 1.171 & 0.622 \\ -0.1037 & 0.4147 & 2.689 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \end{bmatrix} x \end{aligned} \tag{5.89}$$

We need another transformation of the system to ensure $C = \begin{bmatrix} I & 0 \end{bmatrix}$. *From the definition of the extended dynamical structure function, we know that this transformation should be of the form (2.18). For the system in (5.89), this generates the transformation*

$$\bar{T} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 1 & -2 & 1 \end{bmatrix}$$

Applying the transformation to the system in (5.89) yields the state space system:

$$\begin{aligned} \dot{\bar{z}} &= \begin{bmatrix} -10.83 & 6.333 & -2.205 \\ -12.83 & 7.333 & -2.205 \\ 0.5669 & -0.378 & -2.5 \end{bmatrix} \bar{z} + \begin{bmatrix} 1 & 4 & 9 \\ 2 & 6 & 12 \\ 0.378 & -1.512 & 1.134 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \bar{z} \end{aligned} \tag{5.90}$$

which has a corresponding dynamical structure function

$$\bar{Q}(s) = \begin{bmatrix} 0 & \frac{6.333s^2+32.5s+41.67}{s^3+15.83s^2+61.67s+70.83} \\ \frac{-12.83s^2-65.42s-83.33}{s^3-2.333s^2-31.25s-47.92} & 0 \end{bmatrix} \tag{5.91}$$

$$\bar{P}(s) = \begin{bmatrix} \frac{s^2+4.167s+4.167}{s^3+15.83s^2+61.67s+70.83} & \frac{4s^2+23.33s+33.33}{s^3+15.83s^2+61.67s+70.83} & \frac{9s^2+42.5s+50}{s^3+15.83s^2+61.67s+70.83} \\ \frac{2s^2+9.167s+10.42}{s^3-2.333s^2-31.25s-47.92} & \frac{6s^2+33.33s+45.83}{s^3-2.333s^2-31.25s-47.92} & \frac{12s^2+57.5s+68.75}{s^3-2.333s^2-31.25s-47.92} \end{bmatrix}$$

Therefore, we have $(Q(s), P(s)) = (\bar{Q}(s), \bar{P}(s))$ *with* $\hat{T}^{-1}\hat{T}^{-T} = I + \bar{E}$*, where* $\bar{E} \in \mathscr{N}(C)$ *and* $\bar{E} = \bar{E}^T$*.*

5.2.4 Step 4: Pad Realizations with Additional States

Given the transformations that maintain each of the three representations, we are now ready to discuss how to make their corresponding set of representations comparable by ensuring they are all the same order. Given the relationships in Theorem 9, this is accomplished by padding the states of the minimal realizations of the transfer function and dynamical structure function to match the order of any minimal realization of the structured linear fractional transformation.

Transfer Function

In Lemma 9 we discussed the fact that the poles of the transfer function come from the observable and controllable states in a system. That means that the additional states required must be unobservable and/or uncontrollable, or the transfer function of the system would change.

Dynamical Structure Function

The type of additional states used to increase the order of the associated set of realizations depends on whether shared hidden states appear in the system. If no shared hidden states

occur, then no additional states are required since the minimal state space realizations of the structured linear fractional transformation and dynamical structure function will be the same order. However, if there is at least one shared hidden state in the system, then there are two ways to append states:

1. Add unobservable states, these will not show up in the dynamical structure function, or
2. Copy each shared hidden state into its own state. Each of these will be observable, but the underlying dynamical structure function does not change.

Figure 5.4 illustrates the process of appending states to the semantics of both the transfer function and dynamical structure function in order to create a notion of semantics for each that is comparable to the semantics of the structured linear fractional transformation.

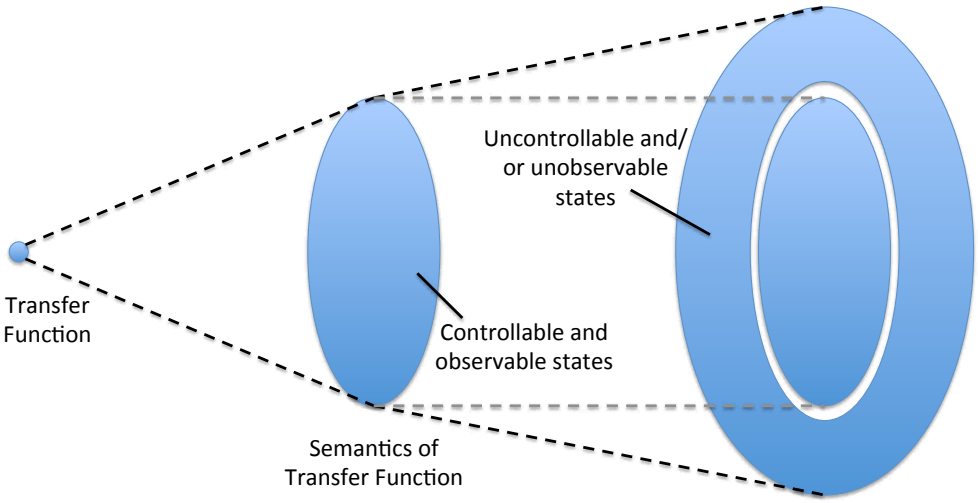
5.2.5 Step 5: Comparison of Networks Semantics

Consider the three transformations that maintain their respective system representations.

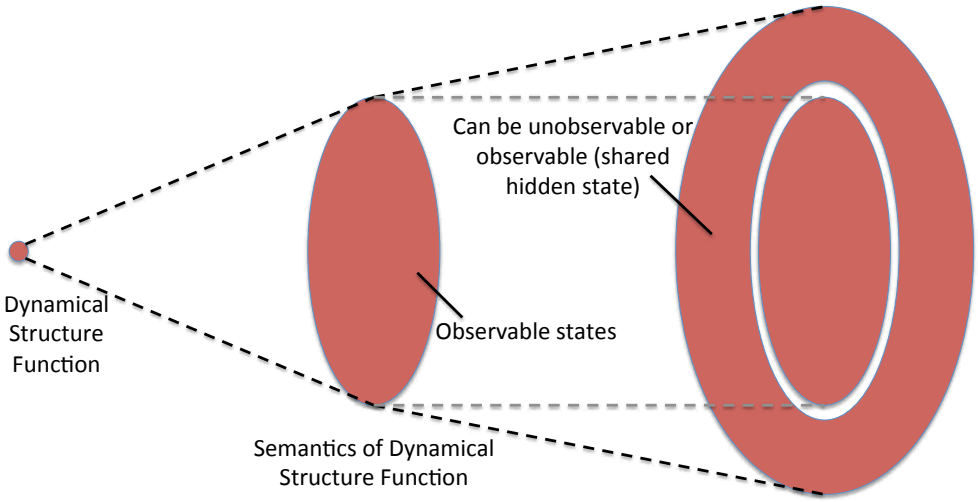
1. In Section 5.2.3 we demonstrated that the transfer function of a system is maintained by any invertible matrix $T \in \mathbb{R}^{n \times n}$. We define the set of all transformations that maintain the transfer function as χ_{tf}.
2. Theorem 10 showed that the structured linear fractional transformation is maintained by any block diagonal, invertible matrix

$$\bar{T} = \begin{bmatrix} T_1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & T_q \end{bmatrix}$$

   We define the set of all transformations that maintain the structured linear fractional transformation as χ_{sub}.



(a) The set of realizations associated with each transfer function, i.e. the semantics of the transfer function, and the allowable states that can be appended without changing the transfer function.



(b) The set of realizations associated with each dynamical structure function, i.e. the semantics of the transfer function, and the allowable states that can be appended without changing the dynamical structure function.

Figure 5.4: Appending states to the ensure comparable semantics.

3. The dynamical structure function can be maintained by any invertible matrix of the form $\hat{T}^{-1}\hat{T}^{-T} = I + \bar{E}$ where $\bar{E} = \bar{E}^T$ and $range(\bar{E}) \subseteq \mathscr{N}(C)$. We define the set of all transformations that maintain the dynamical structure function as χ_{dsf}.

Note that the transformations discussed here work only for the minimal orders of the corresponding system representations. Once we have padded states, the transformations alone do not always maintain the associated representations. The transfer function is always maintained with an invertible transformation T regardless of the order of the system.

As for the transformations on padded realizations that maintain the dynamical structure function, there are two possible cases:

1. Padding realizations so that each shared hidden state is copied allows for transformations that have $\hat{T}^{-1}\hat{T}^{-T} = I + \bar{E}$ where $\bar{E} = \bar{E}^T$ and $range(\bar{E}) \subseteq \mathscr{N}(C)$ where the transformation is of appropriate dimension.
2. Determining a set of properties associated with transformations that maintain the dynamical structure function on realizations that are padded with unobservable states is an open problem. A sufficient condition is for the transformation to be block diagonal with two blocks: one block for the original states, T_1, has the properties already described for maintaining the dynamical structure function and a second block for the padded states, T_2, which must be invertible.

Theorem 12. *Given χ_{tf}, χ_{sub}, and χ_{dsf}, the following relationships hold when the orders of the realizations are the same:*

1. $\chi_{sub} \subseteq \chi_{tf}$
2. $\chi_{dsf} \subseteq \chi_{tf}$
3. $\chi_{dsf} - \chi_{sub} \neq \emptyset$, $\chi_{sub} - \chi_{dsf} \neq \emptyset$, *and* $\chi_{dsf} \cap \chi_{sub} \neq \emptyset$

Proof. First, we will show that $\chi_{sub} \subseteq \chi_{tf}$. Consider an $x \in \chi_{sub}$, since x is invertible we know $x \in \chi_{tf}$. Therefore, $\chi_{sub} \subseteq \chi_{tf}$. Note that set equality occurs when there is only a single subsystem.

Next, we will show that $\chi_{dsf} \subseteq \chi_{tf}$. Consider an $x \in \chi_{dsf}$, since x is invertible we know $x \in \chi_{tf}$. Therefore, $\chi_{dsf} \subseteq \chi_{tf}$. Note that set equality occurs when there is only a single state in the system.

Finally, we will break up the last piece of the theorem into its three parts:

1. We want to show that $\chi_{dsf} - \chi_{sub} \neq \emptyset$, i.e. $(\exists x | x \in \chi_{dsf} \wedge x \notin \chi_{sub})$. Let x be any orthogonal matrix that is not block diagonal, then a transformation of this form maintains the dynamical structure function, but not the structured linear fractional transformation.

2. We want to show that $\chi_{sub} - \chi_{dsf} \neq \emptyset$, i.e. $(\exists x | x \in \chi_{sub} \wedge x \notin \chi_{dsf})$. Let x be any block diagonal matrix that does not meet the conditions $x^{-1}x^{-T} = I + \bar{E}$ where $\bar{E} = \bar{E}^T$ and $range(\bar{E}) \subseteq \mathscr{N}(C)$. Then a transformation of this form maintains the structured linear fractional transformation, but not the dynamical structure function.

3. We want to show that $\chi_{dsf} \cap \chi_{sub} \neq \emptyset$, i.e. $(\exists x | x \in \chi_{sub} \wedge x \in \chi_{dsf})$. Let x be any block diagonal matrix such that $x^{-1}x^{-T} = I + \bar{E}$ where $\bar{E} = \bar{E}^T$ and $range(\bar{E}) \subseteq \mathscr{N}(C)$, then a transformation of this form maintains both the structured linear fractional transformation and the dynamical structure function.

□

The semantic relationship between the three system representations is given in Figure 5.5.

Example 12. *Consider the dynamical structure function:*

$$Q(s) = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{s^2+7s+12} & 0 & 0 \\ \frac{1}{s^2+7s+12} & 0 & 0 \end{bmatrix}, \qquad P(s) = \begin{bmatrix} \frac{1}{s+1} \\ 0 \\ 0 \end{bmatrix} \tag{5.92}$$

The corresponding signal structure of the dynamical structure function in (5.92) is given by:

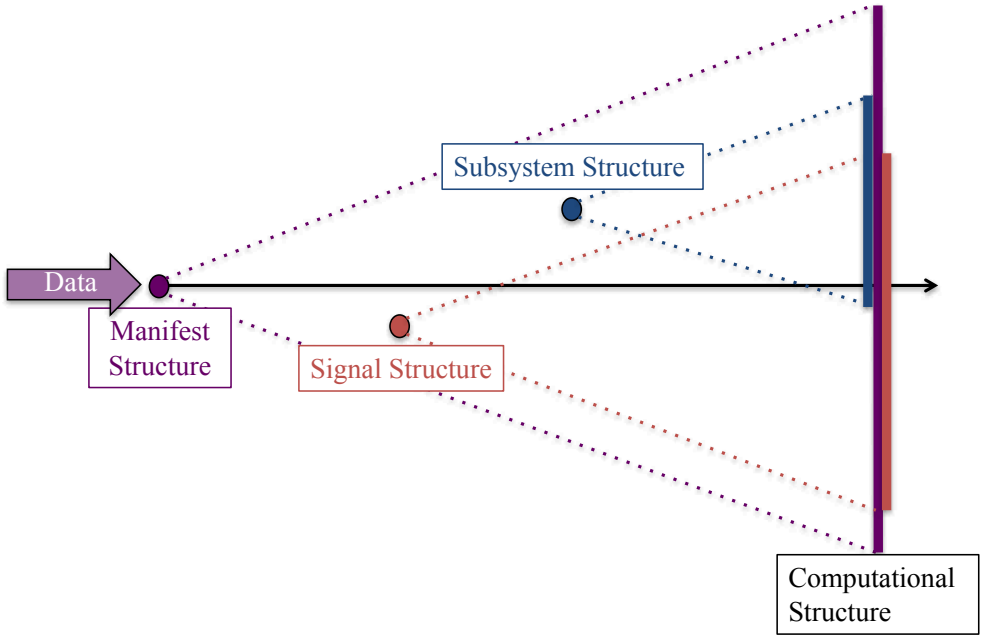


Figure 5.5: The actual relationships between the semantics of the transfer function, structured linear fractional transformation, and the dynamical structure function.

A minimal realization of the dynamical structure function (5.92) is given by:

$$\begin{aligned} \dot{x} &= \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -3 & 0 & 1 \\ 0 & 0 & -3 & 1 \\ 1 & 0 & 0 & -4 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x \end{aligned} \tag{5.93}$$

which is a 4^{th} order state space model.

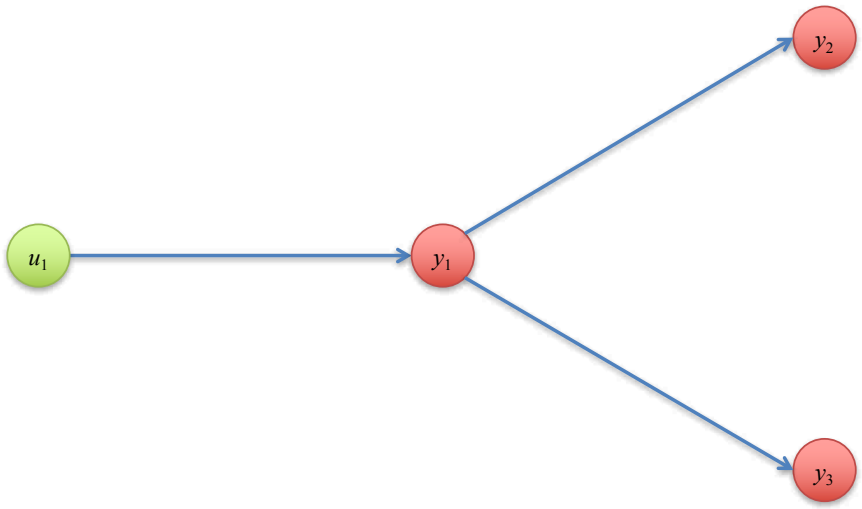


Figure 5.6: The signal structure of the dynamical structure function in (5.92).

1. *The first step in comparing the semantics of various system representations, given a dynamical structure function* $(Q(s), P(s))$ *as in (5.92), is to calculate the associated transfer function. This is accomplished by using the relationship:*

$$G(s) = (I - Q(s))^{-1} P(s)$$

to(5.92), which yields:

$$G(s) = \begin{bmatrix} \frac{1}{s+1} \\ \frac{1}{s^3+8s^2+19s+12} \\ \frac{1}{s^3+8s^2+19s+12} \end{bmatrix} \tag{5.94}$$

The associated manifest structure of the transfer function in (5.94) is given by:

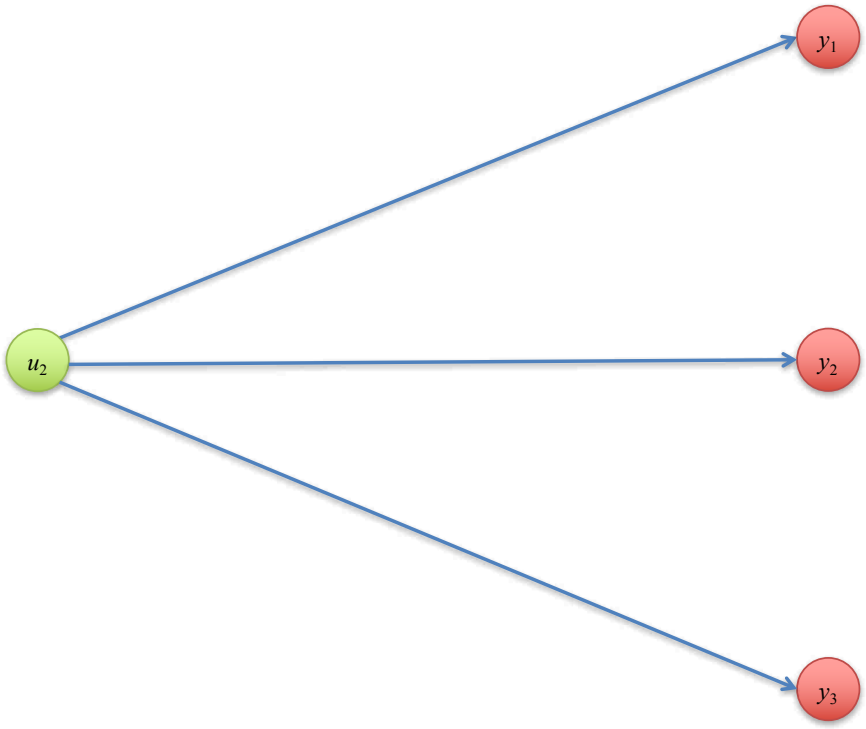


Figure 5.7: The manifest structure of the transfer function in (5.94).

A minimal realization of the transfer function (5.94) is given by:

$$\begin{aligned} \dot{x} &= \begin{bmatrix} -2 & -1 & 1 \\ -\frac{\sqrt{2}}{2} & -4 & -\frac{\sqrt{2}}{2} \\ 1 & 1 & -2 \end{bmatrix} x + \begin{bmatrix} \frac{\sqrt{2}}{2} \\ 0 \\ \frac{\sqrt{2}}{2} \end{bmatrix} u \\ y &= \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ \frac{1}{2} & 0 & -\frac{1}{2} \end{bmatrix} x \end{aligned} \tag{5.95}$$

which is a 3^{rd} order state space model.

2. *The second step is to determine the structured linear fractional transformation given by finding the graphical dual of the signal structure in Figure 5.6, to get a subsystem structure:*

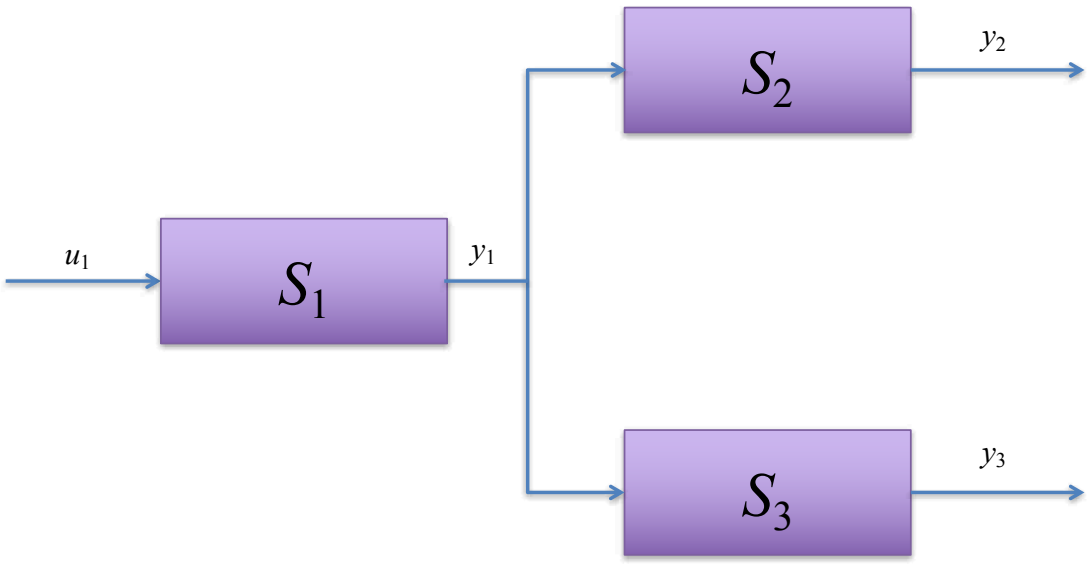


Figure 5.8: The subsystem structure that is the graphical dual of the signal structure in (5.6).

Given the subsystem structure in Figure 5.8, the associated structured linear fractional transformation is:

$$N = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \qquad S(s) = \begin{bmatrix} \frac{1}{s+1} & 0 & 0 \\ 0 & \frac{1}{s^2+7s+12} & 0 \\ 0 & 0 & \frac{1}{s^2+7s+12} \end{bmatrix} \tag{5.96}$$

A minimal realization of the structured linear fractional transformation in (5.96) is

$$\begin{aligned} \dot{x} &= \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 1 & -7 & -12 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -7 & -12 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} x \end{aligned} \tag{5.97}$$

which is a 5^{th} order state space model.

3. *The third step is to determine the associated semantics of each representation. The semantics of the transfer function is any invertible $T \in \mathbb{R}^{3\times 3}$. The semantics of the structured linear fractional transformation is any block diagonal transformation of the*

form

$$\hat{T} = \begin{bmatrix} t_1 & 0 & 0 \\ 0 & T_2 & 0 \\ 0 & 0 & T_3 \end{bmatrix}$$

where $t_1 \in \mathbb{R}^{1\times 1}$, $T_2 \in \mathbb{R}^{2\times 2}$, *and* $T_3 \in \mathbb{R}^{2\times 2}$, *meaning* $\hat{T} \in \mathbb{R}^{5\times 5}$. *Finally, we utilize the transformation* $\bar{T} \in \mathbb{R}^{4\times 4}$ *where* $\bar{T}\bar{T}^{-T} = I + \bar{E}$ *such that* $I + \bar{E}$ *where* $range(\bar{E}) \subseteq \mathscr{N}(C)$ *and* $\bar{E} = \bar{E}^T$.

4. *In order to make all the semantics comparable, we pad the states of the minimal realization of the transfer function and the dynamical structure function. For the transfer function we need to add two states that are uncontrollable and/or unobservable. For the dynamical structure function we can add an unobservable state or we can use the minimal realization of the structured linear fractional transformation to describe the semantics.*

5. *The final step is the comparison of the corresponding sets. Figure 5.9 demonstrates the sets of state space realizations associated with each system representation for different orders of realizations. Note that for* 3^{rd} *order systems the only semantics available are for the transfer function. As we move onto* 4^{th} *order systems, we continue to have semantics for transfer functions, but now we also include dynamical structure functions. Finally, for* 5^{th} *order system, we have the semantics for the transfer function, dynamical structure function, and structured linear fractional transformation.*

   Consider the set of all 3^{rd} *order state space models. Since the transfer function is the only representation that is associated with a realization of order 3, it is the only representation with a set of semantics in that universe. The set, which we denote* χ^3_{tf} *is characterized by a* 3×3 *invertible matrix, which means it has* 9 *free parameters.*

   If we move onto 4^{th} *order state space models, we now have semantics for both the transfer function,* χ^4_{tf}, *and the dynamical structure function,* χ^4_{dsf}. *Since* χ^4_{tf} *is the set*

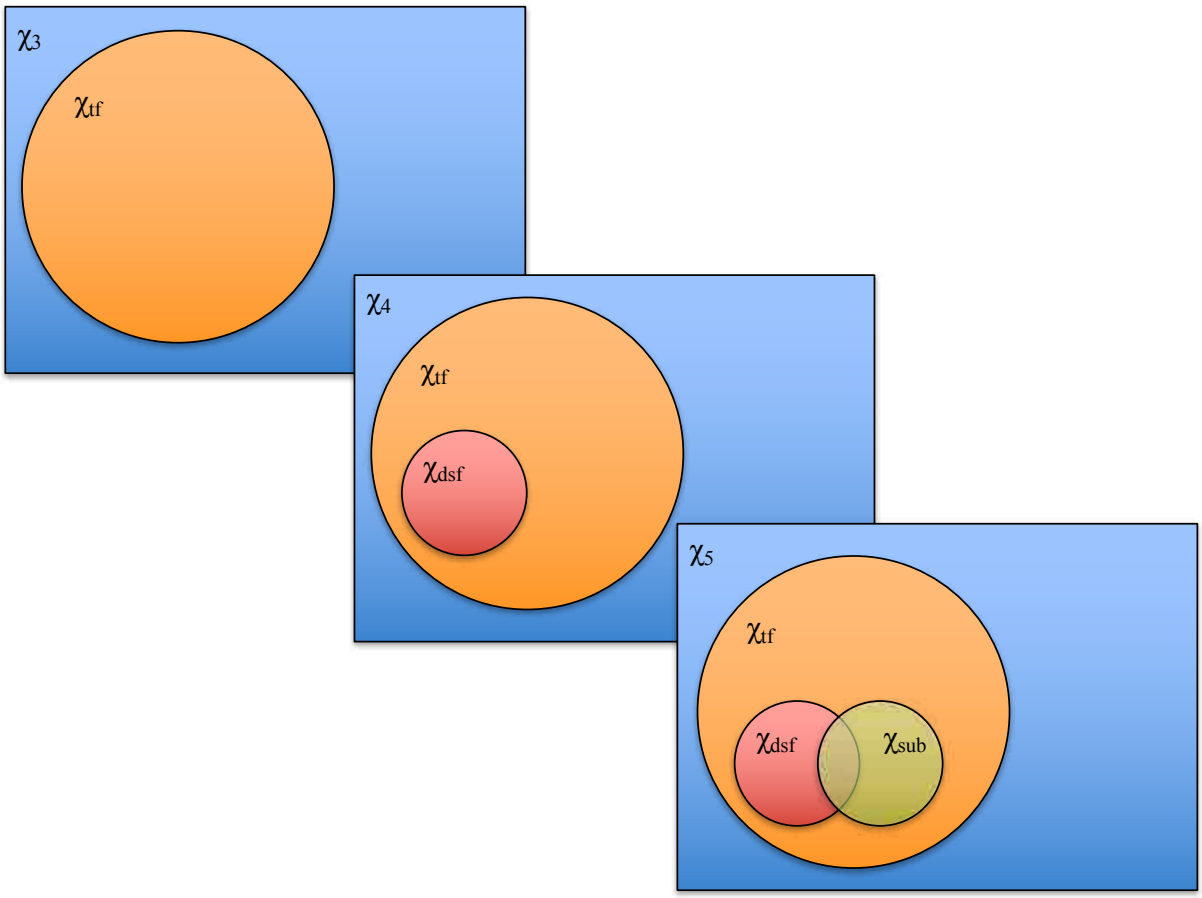


Figure 5.9: The relationship of semantics across different state space orders of the three system representations: the transfer function, the dynamical structure function, and the structured linear fractional transformation.

of all invertible matrices of size 4×4*, the transformation* T *has 16 free parameters and maps* $x_1 \in \chi^4_{tf}$ *to* $x_2 \in \chi^4_{tf}$*. Another transformation* $\bar{T}$ *maps* $\bar{x}_1 \in \chi^4_{dsf}$ *to* $\bar{x}_2 \in \chi^4_{dsf}$*; determining the number of free parameters is still an open problem, since the only conditions we have on* $\bar{T}$ *is* $\bar{T}\bar{T}^{-T} = I + \bar{E}$ *where range*$(\bar{E}) \subseteq \mathcal{N}(C)$ *and* $\bar{E} = \bar{E}^T$*. However, we can show that there exist transformations that maintain the transfer*

function and not the dynamical structure function. For example, the transformation

$$T = \begin{bmatrix} 0.8147 & 0.6324 & 0.9575 & 0.9572 \\ 0.9058 & 0.0975 & 0.9649 & 0.4854 \\ 0.1270 & 0.2785 & 0.1576 & 0.8003 \\ 0.9134 & 0.5469 & 0.9706 & 0.1419 \end{bmatrix} \tag{5.98}$$

maintains the transfer function, but not the dynamical structure function while the transformation

$$\bar{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \sqrt{2} \end{bmatrix} \tag{5.99}$$

maintains both, meaning that $\chi^4_{dsf} \subset \chi^4_{tf}$ *as shown in Figure 5.9.*

Finally, we look at the set of all 5^{th} *order systems. Assuming we pad the set of realizations consistent with the transfer function with uncontrollable and/or unobservable states, the semantics of the systems show us that any invertible transformation, such as*

$$T = \begin{bmatrix} 0.2769 & 0.3171 & 0.7655 & 0.6463 & 0.6551 \\ 0.0462 & 0.9502 & 0.7952 & 0.7094 & 0.1626 \\ 0.0971 & 0.0344 & 0.1869 & 0.7547 & 0.1190 \\ 0.8235 & 0.4387 & 0.4898 & 0.2760 & 0.4984 \\ 0.6948 & 0.3816 & 0.4456 & 0.6797 & 0.9597 \end{bmatrix} \tag{5.100}$$

maintains the transfer function, but not the dynamical structure function or the structured linear fractional transformation.

Next, we will look at the case where we pad the states of the dynamical structure function. One method is to copy the shared hidden state in the same way it is used

for the structured linear fractional transformation. Consider the state space model in (5.97) which is a minimal realization of the structured linear fractional transformation; the associated dynamical structure function is given by (5.92). So although it is not a minimal realization of the dynamical structure function, it is a realization consistent with the dynamical structure function. This means that the state space in (5.97), which we denote x_3, is an element of χ^5_{dsf}, as well as χ^5_{sub} and χ^5_{tf}. This implies that the three sets overlap.

The transformation

$$\bar{T} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{5.101}$$

maps to the state space model

$$\begin{aligned} \dot{z} &= \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & -12 & 0 & -7 & 0 \\ 1 & 0 & -12 & 0 & -7 \end{bmatrix} z + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} z \end{aligned} \tag{5.102}$$

which maintains the dynamical structure function and the transfer function, but not the structured linear fractional transformation. However, the transformation

$$\hat{T} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & -2 & 1 \\ 0 & 0 & 0 & 1 & -2 \end{bmatrix} \tag{5.103}$$

maintains the structured linear fractional transformation and the transfer function, but not the dynamical structure function. Thus, we have

(a) $\chi_{sub} \subseteq \chi_{tf}$

(b) $\chi_{dsf} \subseteq \chi_{tf}$

(c) $\chi_{dsf} - \chi_{sub} \neq \emptyset$, $\chi_{sub} - \chi_{dsf} \neq \emptyset$, *and* $\chi_{dsf} \cap \chi_{sub} \neq \emptyset$

as proven in Theorem 12.

In addition, another way to pad states in the dynamical structure function is to add unobservable states to the 4^{th} *order state space system in (5.93) which leads to:*

$$\begin{aligned} \dot{x} &= \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -3 & 0 & 1 & 0 \\ 0 & 0 & -3 & 1 & 0 \\ 1 & 0 & 0 & -4 & 0 \\ 0 & 0 & 0 & 0 & -5 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} x \end{aligned} \tag{5.104}$$

which is an element of χ_{tf} *and* χ_{dsf}*, but not* χ_{sub}*.*

5.3 Necessary and Sufficient Conditions for Reconstruction of the Structured Linear Fractional Transformation

Consider the relationship between the transfer function and structured linear fractional transformation

$$G(s) = (I - S(s)K)^{-1}S(s)L.$$

This can be rearranged as

$$G(s) = S(s)L + S(s)KG(s) \tag{5.105}$$

$$G(s)^T = (S(s)L)^T + G(s)^T(S(s)K)^T \tag{5.106}$$

$$G(s)^T = \begin{bmatrix} I & G(s)^T \end{bmatrix} \begin{bmatrix} (S(s)L)^T \\ (S(s)K)^T \end{bmatrix} \tag{5.107}$$

Before proceeding, we require the following definition:

Definition 22. *Given* $A \in \mathbb{R}^{m\times n}$ *and* $B \in \mathbb{R}^{p\times q}$*, the* Kronecker product $A \otimes B$ *is the* $mp \times nq$ *matrix:*

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix} \tag{5.108}$$

Given the definition of a Kronecker product, we can show that $\vec{z}$ is the vector created by stacking the columns of Z^T, which means that

$$\vec{g} = \begin{bmatrix} I \otimes I & I \otimes G(s)^T \end{bmatrix} \begin{bmatrix} \vec{sl} \\ \vec{sk} \end{bmatrix} \tag{5.109}$$

Let $F = \begin{bmatrix} I \otimes I & I \otimes G(s)^T \end{bmatrix}$, then $F \in \mathbb{C}^{pm \times pm+p^2}$.

Note that this is in the form $Ax = b$, where A and b are known and x is the unknown. However every entry of $\vec{sl}$ and $\vec{sk}$ are the sum of entries of $S(s)$ multiplied by entries of L and K respectively. Therefore, the next step we take is to separate these sums out. However, before we proceed, we first note that:

1. $S(s) \in \mathbb{C}^{p \times r}$, meaning there are rp unknowns in $S(s)$
2. $L \in \{0, 1\}^{r \times m}$, meaning there are rm unknowns in L
3. $K \in \{0, 1\}^{r \times p}$, meaning there are rp unknowns in K
4. Overall, there are $2rp + rm$ unknowns.

We will call the parameters in $S(s)$ the dynamic parameters, of which there are rp, and the parameters of L and K the structural parameters, of which there are $rp + rm$. Now, note that the parameters of $S(s)$ and $\begin{bmatrix} L & K \end{bmatrix}$ cannot be separated; therefore, we introduce the notion of codependent parameters. Since $S(s)$ is $p \times r$ and L is $r \times m$, there are $p \times m$ entries of $S(s)L$ each containing a sum of r entries, yielding pmr codependent parameters. Similarly, $S(s)K$ has p^2r codependent parameters. Therefore, the structured linear fractional transformation has $pmr + p^2r$ codependent parameters.

Given this, we can rewrite the equation for F as:

$$\vec{g} = \begin{bmatrix} I \otimes (I \otimes 1_{1\times r}) & I \otimes (G(s)^T \otimes 1_{1\times r}) \end{bmatrix} \begin{bmatrix} sl_1 \\ \vdots \\ sl_{pmr} \\ sk_1 \\ \vdots \\ sk_{p^2r} \end{bmatrix}$$

where sl_i is the i^{th} codependent parameter of $S(s)L$ and sk_j is the j^{th} codependent parameter of $S(s)K$. Let $H = \begin{bmatrix} I \otimes (I \otimes 1_{1\times r}) & I \otimes (G(s)^T \otimes 1_{1\times r}) \end{bmatrix}$, then $H \in \mathbb{C}^{pm \times pmr+p^2r}$.

Lemma 13. *Consider the $pmr + p^2r \times k$ transformation T such that $\hat{x} = Tz$, where z is an arbitrary vector of k transfer functions and*

$$\hat{x} = \begin{bmatrix} sl_1 \\ \vdots \\ sl_{pmr} \\ sk_1 \\ \vdots \\ sk_{p^2r} \end{bmatrix}.$$

Let

$$\hat{M} = HT, \tag{5.110}$$

then $\hat{M}$ is injective if and only if $k \leq pm$ and $rank(\hat{M}) = k$.

Proof. This stems from the fact that $\hat{M}$ is a $pm \times k$ matrix, and $\hat{M}$ is injective if and only if it has full column rank, meaning $k \leq pm$ and $rank(\hat{M}) = k$. □

Definition 23. *Dynamic and structural parameters are considered* active *if they are nonzero at least once with a corresponding parameter in a codependent parameter.*

Lemma 14. *The codependent parameters can be uniquely determined $\iff$ the active structural and dynamic parameters can be uniquely determined.*

Proof. Given the structural and dynamic parameters, the codependent parameters can be determined uniquely by multiplying the structural and dynamic parameters together. Specifically, given L, K and $S(s)$, we can compute $s_{ij}(s)l_{jh}$ and $s_{ij}(s)k_{jh}$ by multiplication of the corresponding entry.

Now, we look more closely at extracting the entries of L, K and $S(s)$ given the codependent parameters of the structured linear fractional transformation. Since L and K are boolean matrices, any non-zero codependent parameter can be decomposed into its structural

and dynamic parameters. A structural or dynamic parameter r cannot be determined from the codependent parameters when every codependent parameter in which it appears is zero. If this is the case, then it is not an active parameter, which completes the proof. □

Theorem 13. *Given a system characterized by the transfer function $G(s)$, the active parameters of its structured linear fractional transformation $(N, S(s))$ can be identified if and only if*

1. *$\hat{M}$, defined in (5.110), is injective, and*
2. *$\vec{g} \in \mathcal{R}(\hat{M})$*

Proof. The proof follows immediately from the observation that $\hat{M}$ is a mapping from unidentified model parameters to the system's transfer function, i.e. $\hat{M}z = \overleftarrow{g}$. Under these conditions one can solve for z given $G(s)$ and then use the reconstructed codependent parameters to determine the active structural and dynamic parameters as in Lemma 14. □

Example 13. *Given the following transfer function of a system:*

$$G(s) = \begin{bmatrix} \frac{1}{s+1} \\ \frac{1}{(s+1)(s+2)} \end{bmatrix} \tag{5.111}$$

we attempt to find the structured linear fractional transformation of the system:

$$N = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ l_{11} & k_{11} & k_{12} \\ l_{21} & k_{21} & k_{22} \\ l_{31} & k_{31} & k_{32} \\ l_{41} & k_{41} & k_{42} \end{bmatrix} S(s) = \begin{bmatrix} s_{11}(s) & s_{12}(s) & s_{13}(s) & s_{14}(s) \\ s_{21}(s) & s_{22}(s) & s_{23}(s) & s_{24}(s) \end{bmatrix} \tag{5.112}$$

Note that from the shape of matrices in (5.112) that $p = 2$, $m = 1$, and $r = 4$. We get the values for p and m directly from the shape of the transfer function, but it may not be

immediately clear why $r = 4$. Given that $p = 2$ and we assume every output of a subsystem is measured, then there are at most $p = 2$ subsystems. Now, given that each subsystem can be an input to the other and we have $m = 1$ input, the number of inputs each subsystem can have is $p - 1 + m = 2$. This means we have two subsystems with two potential inputs each, yielding $r = p(p - 1 + m) = p^2 - p + pm = 4$.

From (5.112) the vector of unknown codependent paramaters is

$$\hat{x} = \begin{bmatrix} s_{11}(s)l_{11} \\ s_{12}(s)l_{21} \\ s_{13}(s)l_{31} \\ s_{14}(s)l_{41} \\ s_{21}(s)l_{11} \\ s_{22}(s)l_{21} \\ s_{23}(s)l_{31} \\ s_{24}(s)l_{41} \\ s_{11}(s)k_{11} \\ s_{12}(s)k_{21} \\ s_{13}(s)k_{31} \\ s_{14}(s)k_{41} \\ s_{11}(s)k_{12} \\ s_{12}(s)k_{22} \\ s_{13}(s)k_{32} \\ s_{14}(s)k_{42} \\ s_{21}(s)k_{11} \\ s_{22}(s)k_{21} \\ s_{23}(s)k_{31} \\ s_{24}(s)k_{41} \\ s_{21}(s)k_{12} \\ s_{22}(s)k_{22} \\ s_{23}(s)k_{32} \\ s_{24}(s)k_{42} \end{bmatrix}. \tag{5.113}$$

From this we can derive the system of equations of the form $H\hat{x} = \overleftarrow{g}$ where

$$H = \begin{bmatrix} H_1 & H_2 & H_3 \end{bmatrix} \tag{5.114}$$

with

$$H_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$H_2 = \begin{bmatrix} \frac{1}{s+1} & \frac{1}{s+1} & \frac{1}{s+1} & \frac{1}{s+1} & \frac{1}{(s+1)(s+2)} & \frac{1}{(s+1)(s+2)} & \frac{1}{(s+1)(s+2)} & \frac{1}{(s+1)(s+2)} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$H_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{s+1} & \frac{1}{s+1} & \frac{1}{s+1} & \frac{1}{s+1} & \frac{1}{(s+1)(s+2)} & \frac{1}{(s+1)(s+2)} & \frac{1}{(s+1)(s+2)} & \frac{1}{(s+1)(s+2)} \end{bmatrix}.$$

Without additional a priori information about the structure or dynamics of the system, we cannot reconstruct the structured linear fractional transformation.

Now, suppose that we know a priori that the boolean matrix L takes the form:

$$L = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.115}$$

This knowledge reduces the number of unknowns in (5.113) from 24 *codependent parameters to* 18*. This is still not enough for reconstruction, so let us further assume that we also know that boolean matrix K takes the form:*

$$K = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{5.116}$$

This knowledge further reduces the number of unknowns in from 18 *codependent parameters to just* 4*. Surprisingly, this is still not enough a priori information for a unique reconstruction. Finally, assume that we also know*

$$\begin{aligned} s_{21}(s)l_{11} &= 0 \\ s_{13}(s)k_{31} &= 0 \end{aligned} \tag{5.117}$$

which leaves use with 2 *unknowns in (5.113), which is sufficient for network reconstruction.*

The vector of unknowns $\hat{x}$ *can then be decomposed into the form* $\hat{T}\hat{z}$ *where* $\hat{T}$ *is a* 24×2 *binary matrix with a* 1 *only in the* $(1,1)$ *and* $(19,2)$ *entries. Then, we can get*

$$\hat{z} = \begin{bmatrix} s_{11}l_{11} \\ s_{23}k_{31} \end{bmatrix} \tag{5.118}$$

Now, from Theorem 13 we can determine the equation $\hat{M}\hat{z} = \overleftarrow{g}$ *where* $\hat{M} = H\hat{T}$*, given by*

$$\begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{s+1} \end{bmatrix} \begin{bmatrix} s_{11}l_{11} \\ s_{23}k_{31} \end{bmatrix} = \begin{bmatrix} \frac{1}{s+1} \\ \frac{1}{(s+1)(s+2)} \end{bmatrix} \tag{5.119}$$

In this case, $\hat{M}$ *is full rank, so from Theorem 13 we know that the system is reconstructible. Since* $\hat{z} = \hat{M}^{-1}\overleftarrow{g}$ *we obtain the following structured linear fractional transformation:*

$$N = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \textit{and } S(s) = \begin{bmatrix} \frac{1}{s+1} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{s+2} & 0 \end{bmatrix} \tag{5.120}$$

5.3.1 Unknown Codependent Parameters

Example 13 demonstrates the main issue with reconstruction of the structured linear fractional transformation, that the size of each subsystem and number of subsystems are not known a priori. This means we need to reconstruct the system based on the worst case scenario that every subsystem is affected by every other subsystem and every external input. Unless we know r a priori, this means we need to determine $pmr + p^2r$ codependent parameters with an upper bound of $r \leq p^2 - p + pm$, i.e. an upper bound on the number of unknowns in a

structured linear fractional transformation representation is

$$p^4 - p^3 + 2p^3m - p^2m + p^2m^2 \tag{5.121}$$

Lemma 15. *Let the amount of a priori information about the codependent parameters of the structured linear fractional transformation be denoted by the triple* (x_1, x_2, x_3) *where*

1. x_1 *be the number of nonzero codependent parameters, i.e. the structural and dynamic parameters are both nonzero,*
2. x_2 *the number of zero-valued structural parameters, and*
3. x_3 *the number of zero-valued dynamic parameters*

we also define x_4 *to be the number of codependent parameters that include the overlap of elements that have both zero-valued structural and dynamic parameters known a priori. Then the structured linear fractional transformation can be reconstructed if*

$$x_1 + px_2 + (p+m)x_3 - x_4 \geq p^4 - p^3 + 2p^3m - p^2m + p^2m^2 - pm \tag{5.122}$$

and the conditions of Theorem 13 hold.

Proof. From the definition of the codependent parameters we know that x_1 reduces one codependent parameter for each known codependent parameter. Moreover x_2 reduces the number of codependent parameters by p for each one that is known, while x_3 reduces the number of codependent parameters by $p+m$ for each one that is known. Finally, x_4 comes from the principle of inclusion and exclusion.

Since H is a $pm \times (pmr + p^2r)$ transfer function matrix and Theorem 13 requires $\hat{M}$ to be injective, we know we must reduce H to a $pm \times k$ transfer function matrix, where $k \leq pm$. This can only be achieved if the a priori information reduces the codependent parameters so that

$$pmr + p^2r - (x_1 + px_2 + (p+m)x_3 - x_4) \leq pm \tag{5.123}$$

which means

$$x_1 + px_2 + (p+m)x_3 - x_4 \geq pmr + p^2r - pm \tag{5.124}$$

Since we know that $r \leq p^2 - p + pm$, we use the upper bound to state

$$x_1 + px_2 + (p+m)x_3 - x_4 \geq p^4 - p^3 + 2p^3m - p^2m + p^2m^2 - pm \tag{5.125}$$

□

Example 14. *Consider the transfer function in (5.111) where the unknown codependent parameters of the structured linear fractional transformation are given by the vector in (5.113), which has 24 unknown parameters. Example 13 had $p = 2$, $m = 1$, and $r = 4$ (as an upper bound). If we apply these values to the equation in (5.121) we get:*

$$2^4 - 2^3 + 2(2)^3(1) - 2^2(1) + 2^2(1^2) = 24 \tag{5.126}$$

Equation 5.125 from Lemma 15 states that

$$x_1 + px_2 + (p+m)x_3 - x_4 \geq p^4 - p^3 + 2p^3m - p^2m + p^2m^2 - pm$$

i.e.

$$x_1 + 2x_2 + 3x_3 - x_4 \geq 22 \tag{5.127}$$

For Example 13, we know that from (5.115) and (5.116) we had $x_2 = 10$ and from (5.117) we had $x_1 = 2$, with $x_3 = 0$ and $x_4 = 0$. Plugging these values into (5.127) yields:

$$2 + 2(10) = 22 \geq 22$$

which, by Lemma 15, is sufficient to reconstruct the structured linear fractional transformation, as was done in (5.120)

5.3.2 Information Cost Comparison

Given these results, we will know focus on the comparison of the structure linear fractional transformation to the dynamical structure function, which means we want to look at the case when each subsystem is a single input-single output transfer function. In this case, we know that $S(s)$ is a diagonal matrix, which greatly reduces the number of unknown codependent parameters in the system meaning less a priori information is needed to reconstruct.

Theorem 14. *Given that each subsystem in a structured linear fractional transformation $(N, S(s))$ is single-input, single-output, i.e. that $S(s)$ is diagonal, and letting α be the number of codependent parameters that are known a priori, then the conditions for reconstruction are that*

$$\alpha \geq p^2 - p \tag{5.128}$$

and the conditions of Theorem 13 hold.

Proof. Given that $S(s)$ is diagonal, we know $S(s)$ is a $p \times p$ transfer function matrix with p unknown dynamic parameters. Furthermore, L is a $p \times m$ boolean matrix with pm unknowns and K is a $p \times p$ matrix with $p^2 - p$ unknowns. The matrix $S(s)L$ yields pm unknown codependent parameters and the matrix $S(s)K$ yields $p^2 - p$ unknown codependent parameters. Therefore, there are $p^2 - p + pm$ unknown codependent parameters and in order to reduce $\hat{H} \in \mathbb{C}^{pm \times (p^2 - p + pm)}$ so that $\hat{M}$ from Theorem 13 is injective, we require that

$$\alpha \geq p^2 - p + pm - pm \text{ or } \alpha \geq p^2 - p$$

□

The following example presents the methodology of reconstructing the structured linear fractional transformation, assuming that it is the graphical dual of the dynamical structure function. This means we assume each subsystem is SISO and each subsystem does not affect itself through a visible node.

Example 15. *Given the transfer function (5.111), we attempt to find the structured linear fractional transformation of the system:*

$$N = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ l_{11} & k_{11} & k_{12} \\ l_{21} & k_{21} & k_{22} \end{bmatrix} \quad S(s) = \begin{bmatrix} s_{11}(s) & 0 \\ 0 & s_{22}(s) \end{bmatrix} \tag{5.129}$$

The associated codependent parameters are now given by

$$\hat{x} = \begin{bmatrix} s_{11}(s)l_{11} & s_{22}(s)l_{22} & s_{11}(s)k_{11} & s_{11}(s)k_{12} & s_{22}(s)k_{21} & s_{22}(s)k_{22} \end{bmatrix}^T \tag{5.130}$$

Another assumption of the dynamical structure function is that the $Q(s)$ matrix is hollow. This can be achieved in the structured linear fractional transformation by setting $k_{11} = 0$ and $k_{22} = 0$. This gives us the codependent parameters as

$$\hat{x} = \begin{bmatrix} s_{11}(s)l_{11} & s_{22}(s)l_{22} & s_{11}(s)k_{12} & s_{22}(s)k_{21} \end{bmatrix}^T \tag{5.131}$$

which gives us only 4 unknown parameters as opposed to the 24 from Example 13. This demonstrates the dramatic reduction in parameters brought about by the assumption that the structured linear fractional transformation is the dual of the dynamical structure function. Now, we can write $\hat{H}\hat{x} = \overleftarrow{g}$ as

$$\begin{bmatrix} 1 & 0 & \frac{1}{(s+1)} & 0 \\ 0 & 1 & 0 & \frac{1}{(s+1)(s+2)} \end{bmatrix} \begin{bmatrix} s_{11}(s)l_{11} \\ s_{22}(s)l_{22} \\ s_{11}(s)k_{12} \\ s_{22}(s)k_{21} \end{bmatrix} = \begin{bmatrix} \frac{1}{s+1} \\ \frac{1}{(s+1)(s+2)} \end{bmatrix} \tag{5.132}$$

Assume that we know $p^2 - p = 2^2 - 2 = 2$ *codependent parameters a priori, as mentioned in Theorem 14. In particular, assume we know* $s_{22}(s)l_{22} = 0$ *and* $s_{11}(s)k_{12} = 0$*, then we can decompose* $\hat{x}$ *into* $\hat{T}\hat{z}$ *as follows:*

$$\hat{T} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \text{ and } \hat{z} = \begin{bmatrix} s_{11}(s)l_{11} \\ s_{22}(s)k_{21} \end{bmatrix} \tag{5.133}$$

Now, from Theorem 13 we can determine the equation $\hat{M}\hat{z} = \overleftarrow{g}$ *where* $\hat{M} = H\hat{T}$*, given by*

$$\begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{s+1} \end{bmatrix} \begin{bmatrix} s_{11}(s)l_{11} \\ s_{22}(s)k_{21} \end{bmatrix} = \begin{bmatrix} \frac{1}{s+1} \\ \frac{1}{(s+1)(s+2)} \end{bmatrix} \tag{5.134}$$

which yields the structured linear fractional transformation

$$N = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} S(s) = \begin{bmatrix} \frac{1}{s+1} & 0 \\ 0 & \frac{1}{s+2} \end{bmatrix} \tag{5.135}$$

5.4 Conclusion

Note that we detailed three methods for comparing system representations: orders of minimal realization, sets of network semantics, and information cost of identifiability conditions. The order of a minimal realization of a dynamical structure function and its associated information cost for reconstruction are both lower than respective order of a minimal realization and associated information cost of the structured linear fractional transformation. However, the semantics of the system also suggest another relationship, demonstrating that the set of state space realizations associated with the dynamical structure function is different from,

but possibly overlaps with, the set of state space realizations associated with the structured linear fractional transformation. Future work will investigate the relationship between the two partial structure representations further.

Part II

Applications: Chapters 6 - 8

Chapter 6

Polynomial-Time Reconstruction of the Dynamical Structure Function

(Published at CDC 2013 as "Robust Signal-Structure Reconstruction")

Abstract

This paper focuses on the reconstruction of the signal structure of a system in the presence of noise and nonlinearities. Previous results on polynomial time reconstruction in this area were restricted to systems where target specificity was part of the inherent structure, [16]. This work extends these results to all reconstructible systems and proposes a faster reconstruction algorithm along with an improved model selection procedure. Finally, a simulation study then details the performance of this new algorithm on reconstructible systems.

6.1 Introduction

The process of network reconstruction is the attempt to determine the structure and dynamics of a networked system. The simplest representation of a linear time-invariant system is its transfer function, G. The process for determining a system's transfer function from input-output data is known as system identification, (see Figure 6.1). Unfortunately, a system's transfer function contains very little information about the internal structure of a network.

A linear time-invariant system's state space realization offers a more detailed representation of a system's structure. Although rich in information, the process of reconstruction from input-output dynamics to the state space realization, (A, B, C, D), known as the realization process (see Figure 6.1), is ill-posed since there are many possible state space realizations for a single transfer function matrix.

Another representation of the structure of a network is a system's dynamical structure function, (Q, P), which was originally introduced in [24]. Dynamical structure functions (DSF) contain more information about a system's structure than the transfer function, while requiring only weak a priori information, compared to the state space realization, to reconstruct from input-output dynamics.

The DSF describes the network structure of a system in the sense that the matrix Q can be interpreted as the weighted adjacency matrix of a directed graph indicating the causal relationships between measured states. Also, P is the weighted adjacency matrix of a directed graph indicating the causal relationships between inputs and measured states. The weights on the edges of this graph are TFs between relevant variables. This graphical representation of the DSF is referred to as the signal structure of the system.

The DSF of a system denotes the structure and dynamics of a linear time-invariant system at a resolution consistent with the number of measured states. This means if less states are measured, the structure of the DSF would relate closely to the structure of a

system's transfer function, while more measured states implies that the structure of the DSF is closer to the structure given by the state space representation of the system.

As with a system's state space, the process of determining a system's DSF from input-output data is ill-posed without any a priori information about the network. However, given a DSF of a system (Q, P), the transfer function for that system is uniquely defined as $G = (I - Q)^{-1}P$, [24].

Definition 1. *A system's DSF is considered* reconstructible *if there exists a priori information about the network that creates a bijection between a system's transfer function and it's DSF.*

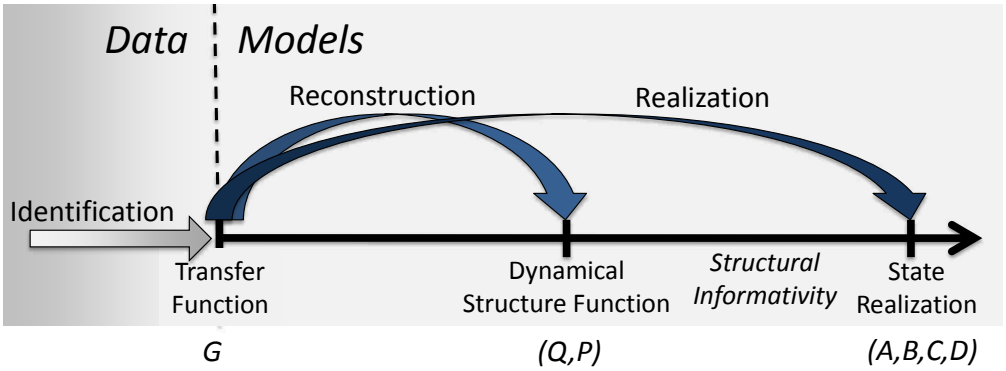


Figure 6.1: System Representations Organized by Structural Informativity

In [73], a robust reconstruction method was presented that allowed for the reconstruction of the signal structure of a network when noise and nonlinearities were present in the system. This approach calculates the optimal dynamical structure function for all possible Boolean structures, i.e. all possible ways of connecting the network, and then uses a model selection technique to determine the best possible Boolean structure. Unfortunately, iterating over all possible Boolean structures involves a computational complexity of $O(2^p)$, which greatly restricts its usage to that of small networks, e.g. networks with less than three or four measured states.

Several algorithms were proposed in [16] that improved the computational complexity of the robust reconstruction method from exponential to polynomial. However, the algorithms proposed are for systems which are target specific, meaning for each measured state there

exists a corresponding input that perturbs that measured state, possibly through a hidden (unmeasured) state, and that input does not perturb any other measured states except through the corresponding measured state. In [7], necessary and sufficient conditions for network reconstruction were proposed that show that target specificity is sufficient, but not necessary for reconstruction. This implies that there exist reconstructible networks for which the target specificity assumption fails. This paper extends the polynomial time algorithm to these cases, which can be common, for example, in proteomics and other applications.

In Section 6.2, we extend the robust reconstruction problem to include all reconstructible networks, not only those that meet the target specificity assumption. In Section 6.3, we extend the polynomial time algorithm for dynamical structure functions to all reconstructible networks and propose a new reconstruction algorithm that reduces the computational complexity of the reconstruction process. Section 6.4 provides an improved model selection procedure. Section 6.5 contains the results of simulation studies. Finally, in Section 6.6 we present our conclusions.

6.2 Robust Reconstruction for Dynamical Structure Functions

Previous robust reconstruction results in [73] use a method that requires target specificity. The results in this paper remove the requirement of target specificity to allow for the robust reconstruction of all possible reconstructible networks, [15].

To model the input-output data with noise and nonlinearities, we begin by considering an additive uncertainty on the control structure P, as seen in Figure 6.2. In this framework, the "true" system is given by $(I - Q)^{-1}(P + \Delta)$, where Δ represents unmodeled dynamics, including noise and nonlinearities. Given this uncertainty, we define the distance from data to a particular Boolean structure to be $||\Delta||$, in an appropriate norm.

Figure 6.2 illustrates the relationship:

$$Y = (I - Q)^{-1}(P + \Delta)U.$$

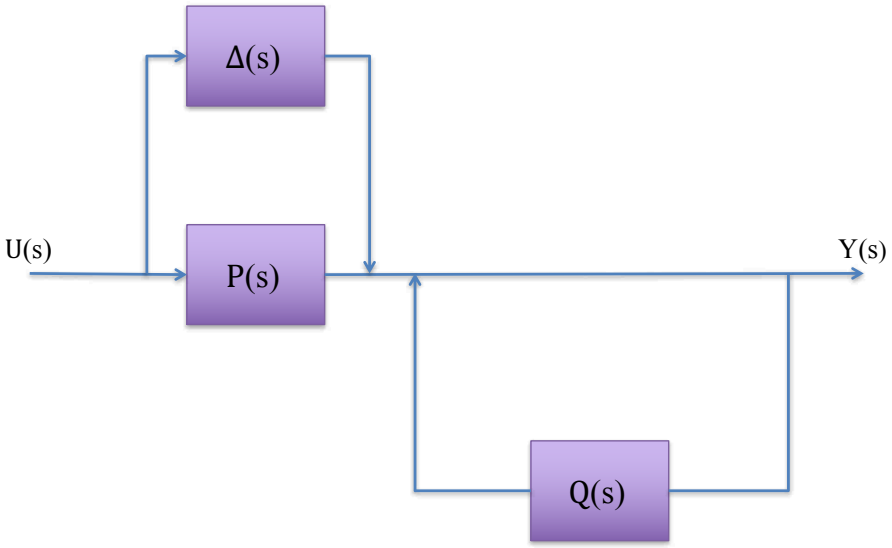


Figure 6.2: Additive uncertainty on P

This can be rewritten as:

$$\Delta U = Y - \begin{bmatrix} Q & P \end{bmatrix} \begin{bmatrix} Y \\ U \end{bmatrix}.$$

We can then exploit the fact that U is a diagonal matrix, based on the way experiments are performed in which only one input is affected at a time, which gives us:

$$||\Delta U|| = ||c\Delta I|| = ||c\Delta|| = |c|||\Delta||,$$

where $c \neq 0$ is the amount each input is perturbed. Thus, we note that minimizing $||\Delta U||$ is equivalent to minimizing $||\Delta||$.

We highlight that this method of determining the correct structure of a network makes the following assumptions:

1. experiments are performed sequentially, with each input taking a turn,
2. the size of the perturbation on each input is of equal magnitude.

The purpose of minimizing $||\Delta||$ is to determine the Boolean structure with the smallest distance from given input-output data from experiments performed on the system. Therefore, we want to solve the following problem:

$$\delta = \min_{Q,P} ||Y - \begin{bmatrix} Q & P \end{bmatrix} \begin{bmatrix} Y \\ U \end{bmatrix} ||$$

Stacking the unknowns from Q and P into a vector x, this can be written as $w - Lx$, where w is the matrix Y stacked into a vector row by row and

$$L = \begin{bmatrix} y_2 \dots y_n & 0 & \dots \dots & 0 & u_1 \dots u_n & 0 & \dots \dots & 0 \\ 0 \dots 0 & \ddots & 0 \dots & 0 & 0 \dots 0 & \ddots & 0 \dots & 0 \\ 0 \dots \dots 0 & y_1 \dots y_{n-1} & 0 & \dots \dots & 0 & u_1 \dots u_n & & \end{bmatrix} \tag{6.1}$$

where y_i and u_i are the i^{th} columns of Y^T and U^T, respectively. Note that we remove a column of y on each row of L because the definition of Q from [24] states that the diagonal values of Q are known to be zero. This means that there are $p^2 - p$ possible Boolean structures of Q and pm possible Boolean structures of P, yielding $pm + p^2 - p$ possible Boolean structures for x.

If we index the possible combinations of Boolean structures with $v = 1, ..., pm + p^2 - p$, then consider the v^{th} Boolean dynamical structure function and denote (Q_v, P_v) as a dynamical structure function with this Boolean structure. We can then reorganize the problem so that it becomes:

$$\delta_v^2 = \inf_{x \in \chi_v} ||w - Lx||_2^2 \tag{6.2}$$

where χ_v is the set of all x that satisfy the constraints of the v^{th} Boolean structure.

It is well known that this problem is ill-posed, since L is not full column rank. As proposed in [7], certain elements of x must be known a priori in order for the system to be reconstructible, this information is contained in a $(pm + p^2 - p) \times k$ transformation matrix T

so that $x = Tz$, where z is the reduced number of unknowns such that LT has full column rank, meaning $k \leq p^2$. The information contained in T could come from knowing how inputs affect the system or how states within the system interact (or fail to interact) with each other. For example, if we knew that the system was target specific, then we know that P is square and diagonal, which is information that can be incorporated into T. The complete necessary and sufficient conditions for reconstructibility are provided in [7].

Therefore, the robust reconstruction problem can finally be stated as:

$$\delta_c^2 = \inf_{z \in \zeta_c} ||w - Mz||_2^2 \tag{6.3}$$

where $M = LT$ and $\zeta_c \subset \chi_v$ is the set of all z that satisfy the constraints of the c^{th} Boolean structure, where $c = 1, ..., k$.

6.3 Polynomial Time Reconstruction Algorithm

Assuming our system meets the requirements to be reconstructible, we now develop a polynomial time algorithm for robust reconstruction. A greedy polynomial-time algorithm for the reconstruction of networks with target specificity was given in [16].

The extension of the algorithm to all reconstructible networks, including those that may fail target specificity, requires the redefinition of several terms in the algorithm. First, we redefine the term S, which originally represented the Boolean structure of Q in [16], to be the Boolean structure of the vector z, which could contain elements of both P and Q. Furthermore, we note that we utilize δ as defined in Equation 6.3 rather than δ as defined in [16]. The superscripts on S and δ in Algorithm M_2 and M_3 refer to the number of links (i.e. non-zero elements) for that Boolean structure, unless otherwise stated.

Algorithm M_2

Set S^k to the fully-connected structure.

for $j = k \to 1$ **do**

Remove one link of S^j at a time to obtain a set of j structures with $j-1$ links and calculate δ^{j-1} for each of these structures.

Set S^{j-1} as the minimum-δ^{j-1} structure.

end for

Set S^0 as the decoupled structure.

Apply a model selection procedure to the set $\mathbb{S} = \{S^j\}$.

We now propose a new algorithm, similar to the one above, which takes advantage of two important facts:

1. Algorithm M_2 is a greedy algorithm that keeps a record of only one structure for each possible structure with j links, where j ranges from 0 to k and
2. As noted in [73], the structures with more links have lower δ scores since they have more degrees of freedom.

The algorithm will also make use of the following definition:

Definition 2. *The term* full *refers to the Boolean structure of the fully-connected network, i.e. a* $k \times 1$ *vector of ones. The term* full-x*, where* x *is a positive integer less than* k*, refers to a Boolean structure with* x *links missing, i.e.* x *entries of* full-x *are zeros, the rest are ones.*

Example 16. *If the* full *Boolean structure of a system is given by* $\begin{bmatrix}1&1&1\end{bmatrix}^T$*, then the possible* full-1 *Boolean structures are:* $\begin{bmatrix}0&1&1\end{bmatrix}^T$, $\begin{bmatrix}1&0&1\end{bmatrix}^T$, *and* $\begin{bmatrix}1&1&0\end{bmatrix}^T$.

First, we use an iterative procedure to determine which links are the least likely to occur in the correct network. The *full*-1 Boolean structure with the j^{th} value set to zero is denoted S_j^1 and its associated δ from Equation 6.3 we denote δ_j^1. Then, another iterative

procedure determines a candidate set $\mathbb{S}$, with one structure (S^k) for each level of sparsity (k links). Finally, a model selection procedure is applied to this reduced set to select a single solution.

Algorithm M_3

Set S^k to the fully-connected structure.

for $j = k \rightarrow 1$ **do**

Set the j^{th} position of S_j^1 to 0 and calculate δ_j^1.

Store S_j^1, δ_j^1, and j in F.

end for

Sort F by δ_j^1 in descending order.

Set S to the fully-connected structure.

for $d = k \rightarrow 1$ **do**

Remove from S the link corresponding to the 0 location of the d^{th} structure from F.

Set S^d to S.

Calculate δ^d for S^d.

end for

Set S^0 as the decoupled structure.

Apply a model selection procedure to the set $\mathbb{S} = \{S^d\}$

Algorithm M_3 reduces the overall number of structures that need to be considered from $O(p^4)$ for Algorithm M_2 to $O(p^2)$, [16].

This means that Method M_3 only needs to consider the structures circled in red in Figure 6.3, rather than all allowable structures considered by Method M_2 in order to determine the correct structure.

To determine when a solution can be found by Algorithm M_3, we begin with the following Lemma:

Lemma 16. *A given reconstruction is solvable by Method M_3, i.e. the true structure will appear in the candidate set, if no* full-1 *link structure with a zero that does not appear in the*

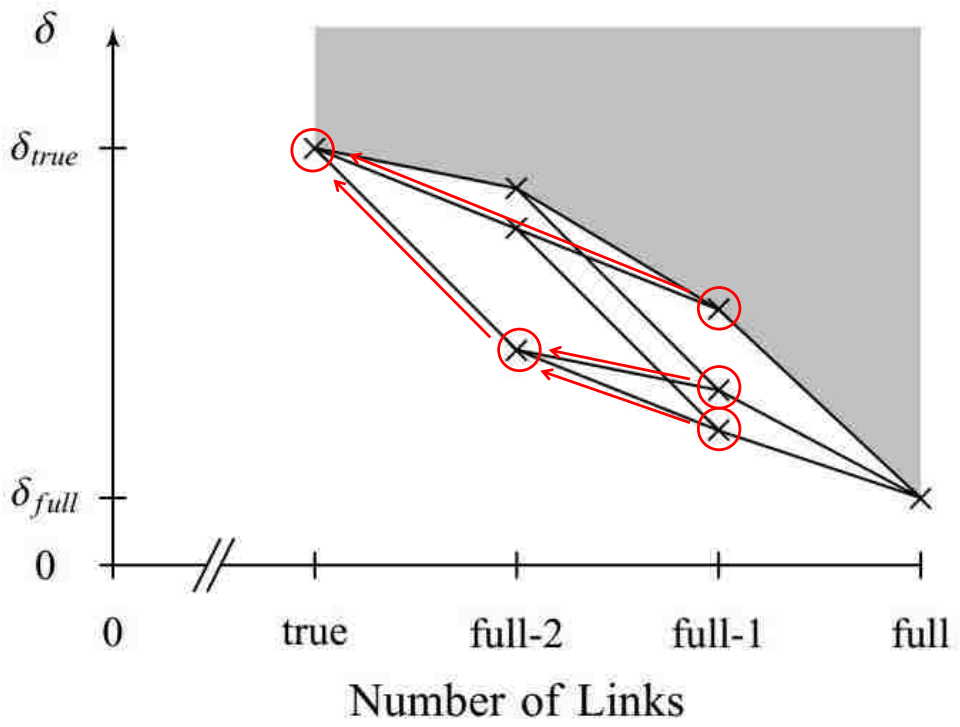


Figure 6.3: An example problem showing δ values plotted against number of links for the allowable set of Method M_2. The structures considered by Method M_3 are circled in red.

true structure has a lower δ score than all full-1 *link structures with zeros that appear in the true structure.*

Proof. If all the *full*-1 link structures that have zeros in the true system have lower δ scores than every other *full*-1 link structure, then their combination will be selected by Method M_3 ensuring that the true solution will appear in the candidate set. □

Theorem 15. *If no noise is present in the system, Method M_3 will ensure that the true structure is part of the candidate set, assuming there is sufficient data for reconstruction.*

Proof. As noted in [73], due to the additional degrees of freedoms provided by extra connections, all Boolean structures S^j with zeros obtained from any ordered combination (where order refers to the iterative manner in which links were eliminated to obtain S^j) which coincide with locations of zeros in the true structure will have a cost $\delta = 0$. Furthermore, all

Boolean structures that have at least one zero that does not correspond to missing links in the true structures will have $\delta > 0$.

Therefore, when no noise is present in the system all Boolean structures with one link missing that correspond to a missing link in the true structure will have δ values lower than all Boolean structures with one link missing that does not appear in the true structure. By Lemma 16, this guarantees that the true structure is part of the candidate set. □

Corollary 3. *There exists $\epsilon > 0$ and $r > 0$, where ϵ represents the noise in the system, such that if $\epsilon < r$, Method M_3 will ensure that the true structure is part of the candidate set, assuming there is sufficient data for reconstruction.*

Proof. This follows from Theorem 15 by continuity. □

6.4 Model Selection Procedure

The original robust reconstruction method in [73] noted that finding an optimal δ yields a series of candidate solutions that have more degrees of freedom than the true network due to overfitting, so a model selection procedure is required to penalize extra connections in the candidate solutions. The Akaike Information Criterion (AIC) was proposed as a model selection procedure and is defined as:

$$AIC = 2k - 2ln(L)$$

where k is the number of parameters in the model and L is the maximized value of the likelihood function for the model, [14].

Akaike's Information Criterion with correction for finite sample sizes is defined as:

$$AICc = AIC + \frac{2k(k+1)}{n-k+1}$$

where n is the sample size.

A customized AIC was used in [73] for the minimization of the likelihood function and was defined as follows:

$$AIC_{original} = 2k + nln\left(\frac{2\pi L}{n} + 1\right) \tag{6.4}$$

However, $AIC_{original}$ did not scale well to large networks and the use of the natural logarithm heavily favored the full link structure when noise was present in the system. To overcome these issues we use a customized form of the Akaike Information Criterion, which we will call the Chetty-Warnick Information Criterion:

$$CWIC_k = \frac{\delta_k}{N + C} + L_k$$

where δ_k is defined in Equation (6.2), N is the number of unknowns in Q, C is the number of unknowns in P, and L_k is the number of nonzero entries in the k^{th} Boolean structure. Dividing δ by $N + C$ scales δ by the size of the known network, since the difference in δ for the candidate solutions becomes smaller as the size of the measured network increases. Furthermore, we use only the minimized value of δ rather than the $ln(\delta)$ because, as mentioned above, the natural logarithm heavily favors the completed connected network, making it difficult to correctly identify the true network.

$CWIC$ with correction for finite sample sizes is then given by:

$$CWICc_k = VIC_k + \frac{2L_k(L_k + 1)}{N + C - L_k + 1}$$

A comparison of the reconstruction process as noise variance increases using Method M_3 with $AIC_{original}$ and the $CWIC$ is given in Figure 6.4. The network being reconstructed is a linearized version of the single feedback loop defined in [73].

As Figure 6.4 shows, $CWIC$ performs better than the original AIC as noise variance increases.

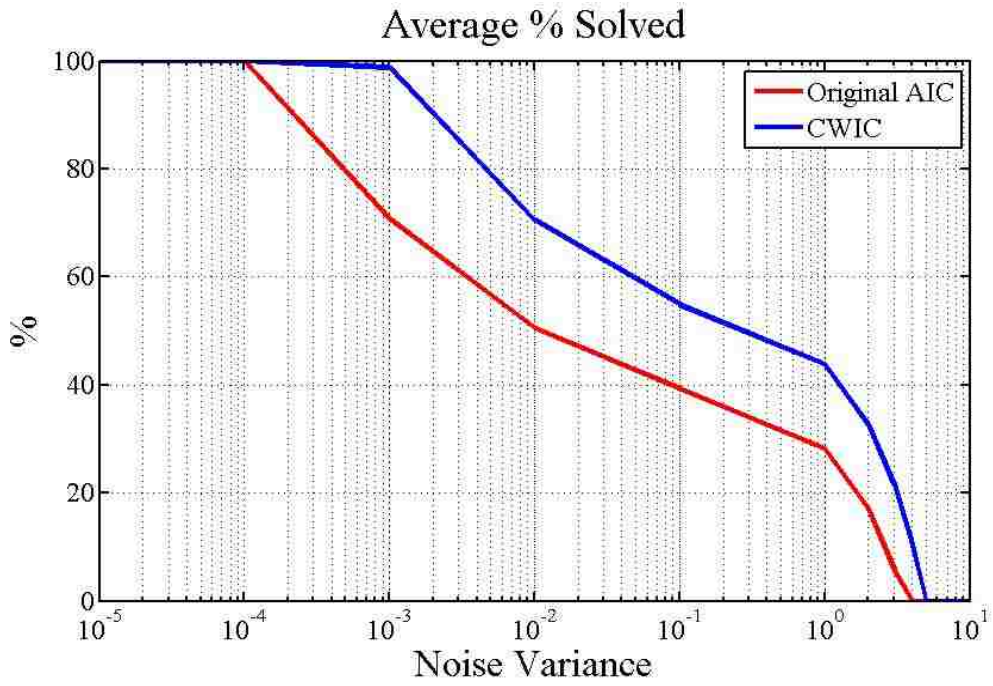


Figure 6.4: Comparison of reconstruction problems successfully solved using Algorithm M_3 using the original AIC and $CWIC$ as the model selection procedure.

6.5 Simulations

Our empirical study will focus on:

1. Compare the accuracy of the Method M_3 before the redefinition of variables from [16] to Method M_3 as defined in Section 6.3 of this paper,
2. Comparing the accuracy of Method M_2 to Method M_3, where the model selection procedure is the $CWIC$,
3. Determining whether the accuracy of reconstruction is degraded by increasing the size of a network, and
4. Improving the accuracy of reconstruction through improved data collection techniques.

The data in these simulations are continuously sampled, with no missing data points.

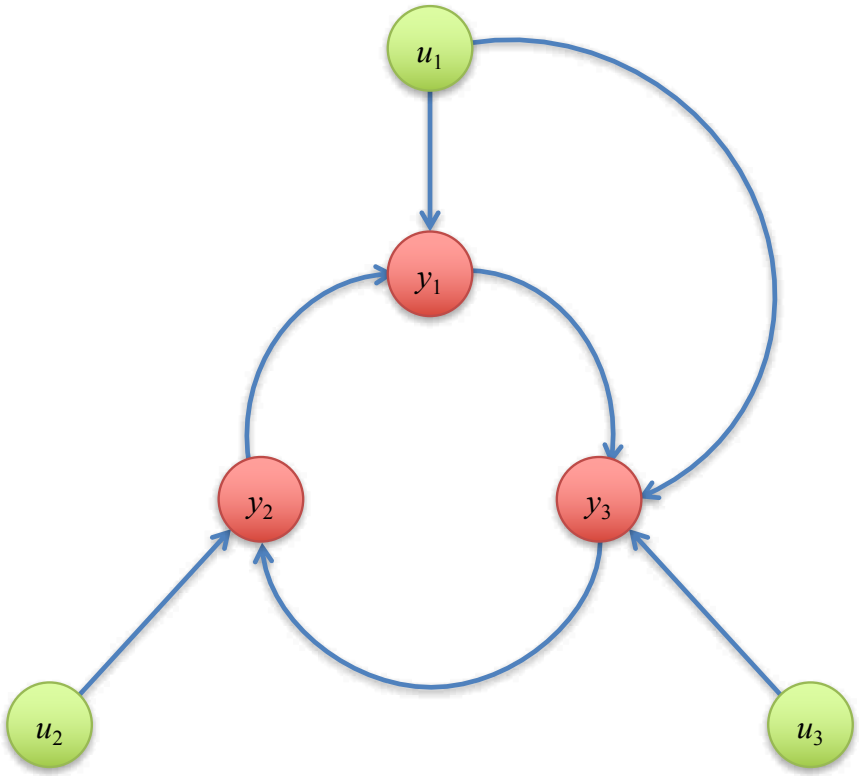


Figure 6.5: Non-Target Specific Ring Network. Green nodes represent inputs and red nodes represent outputs.

6.5.1 Analyzing Non-Target Specific Reconstruction

The first simulation demonstrates why a redefinition of the reconstruction algorithm was even necessary by using a single feedback loop with a single extra edge to make it non-target specific, as seen in Figure 6.5. It is trivial to show that this network is reconstructible.

Figure 6.6 shows the results of reconstruction using Method M_3 both with and without the assumption that the non-target specific network is target specific. As the figure shows, assuming that the system is target specific when it isn't leads to catastrophic failure in the network reconstruction procedure.

6.5.2 Comparison of Polynomial Time Algorithms

Our next result comes from the comparison of Method M_2 and Method M_3. In Figure 6.7, we note that the two methods seem almost indistinguishable.

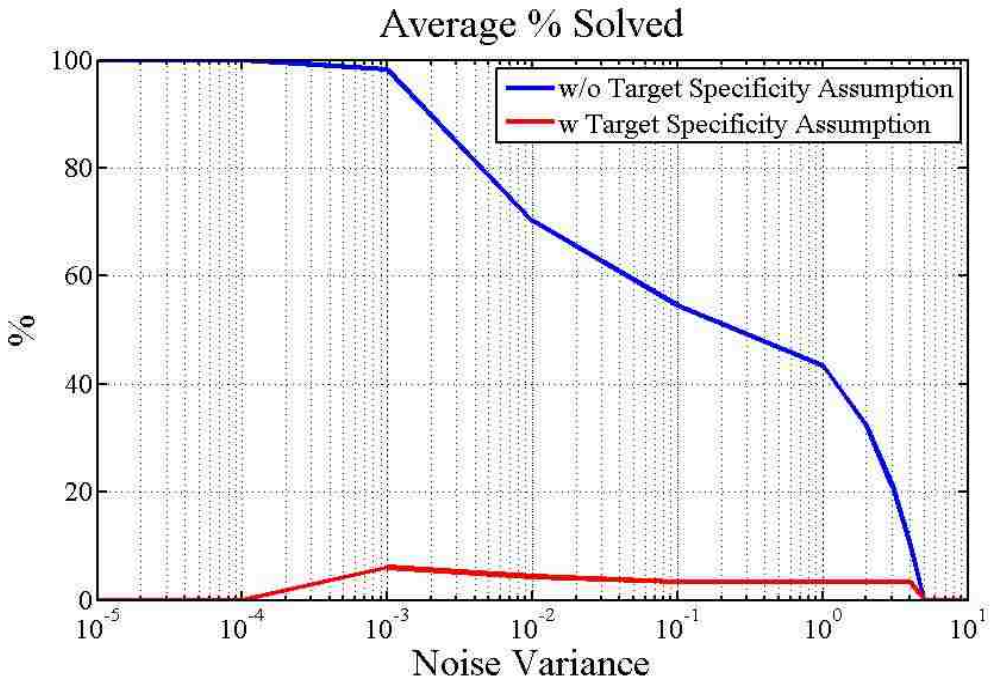


Figure 6.6: Reconstruction of non-target specific network with and without the assumption that it is target specific.

We expect Method M_3 to begin failing when a $full-1$ structure which has a zero that appears in the true structure has a larger δ value than a $full-1$ structure which has a zero that does not appear in the true structure. In this case the true structure is not part of the candidate set provided by Method M_3.

The true structure could potentially be found in the candidate set generated by Method M_2 if the δ value of the structure with non-true zero, mentioned above, combined with zeros that are true values is higher than the combined values of the true zero, mentioned above, with higher δ combined with the other true zero structures.

The fact that the two methods seem almost indistinguishable means that this situation probably doesn't occur for this particular example, although that may not be true in general.

6.5.3 Increasing Network Size

Now, in order to see how the accuracy of reconstruction degrades as the size of the network increases, we introduce the ring of rings network and assume target specificity in Figure 6.8

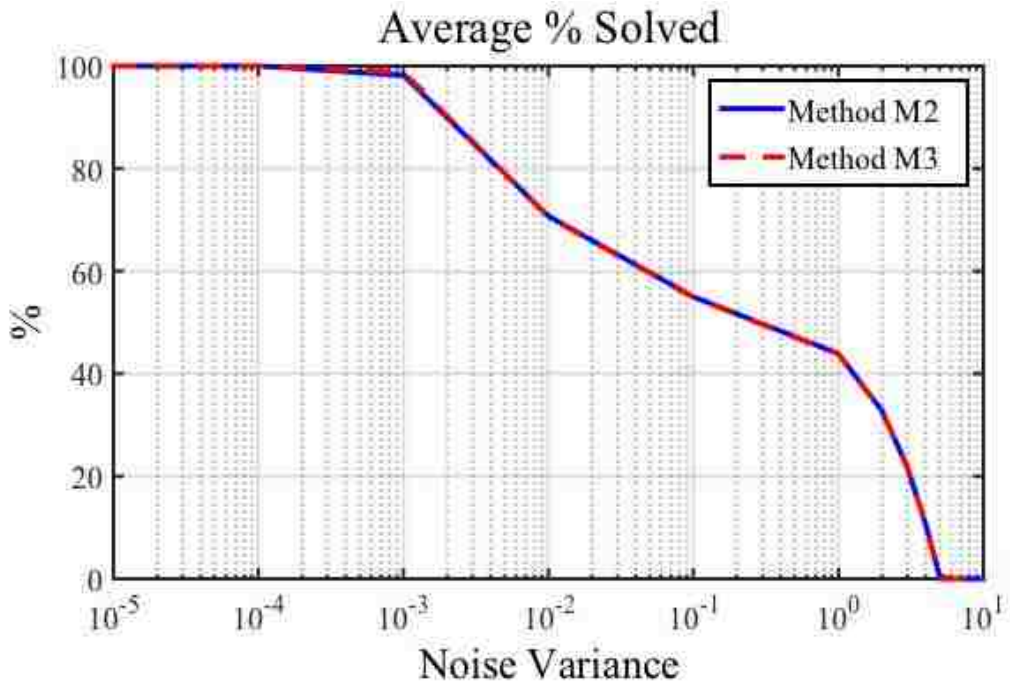


Figure 6.7: Comparison of reconstruction methods M_2 and M_3. Note that the two perform similarly in spite of the fact that M_2 searches over a larger set of possible candidates for the true structure.

(inputs not shown). The rings of rings is interesting since it has a full transfer function, but, as the figure shows, it is clearly very structured.

All things being equal, we now compare the accuracy of reconstructing various subsets of the ring of rings in Figure 6.9. The first set is just a single loop of 3 nodes, the second is two loops of 3 nodes each with one connection between them, the third is three loops of 3 nodes each with a connection between the first and second loop and the second and third loops, and finally the fourth is the complete ring of rings network.

As the figure shows, the network reconstruction process degrades as the size of the network grows. The point of a polynomial time algorithm is to allow for the reconstruction of large networks, so if the reconstruction process degrades with network size, the new algorithm isn't very useful. We now provide several ways in which to improve data collection so that network reconstruction of large networks is viable.

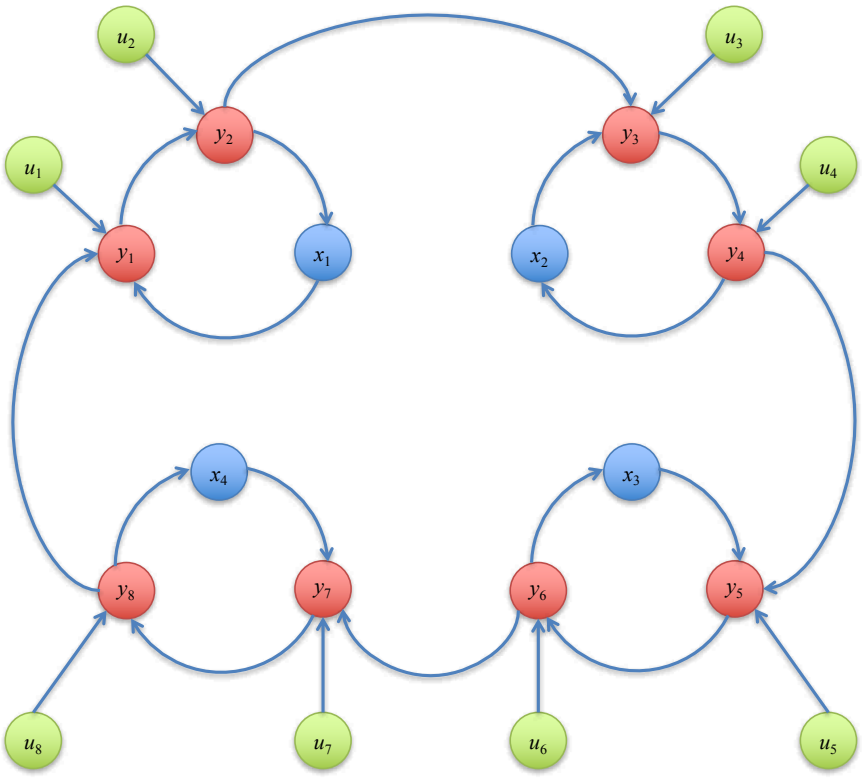


Figure 6.8: Ring of Rings Network

6.5.4 Improving Results

The best way to improve results is to improve data collection.

Repeated Experiments

Firstly, we note that repeating experiments on the network can drastically increase the accuracy of reconstruction by averaging out noise in the system. If the cost of experiments is high, reconstruction is still possible, but with much smaller ϵ. This is made evident in comparing reconstruction for increasing noise variance with no repeated experiments to reconstruction with noise averaging, again using a linearized single feedback loop, in Figure 6.10.

The figure shows that increasing the noise averaging allows for more accurate reconstruction.

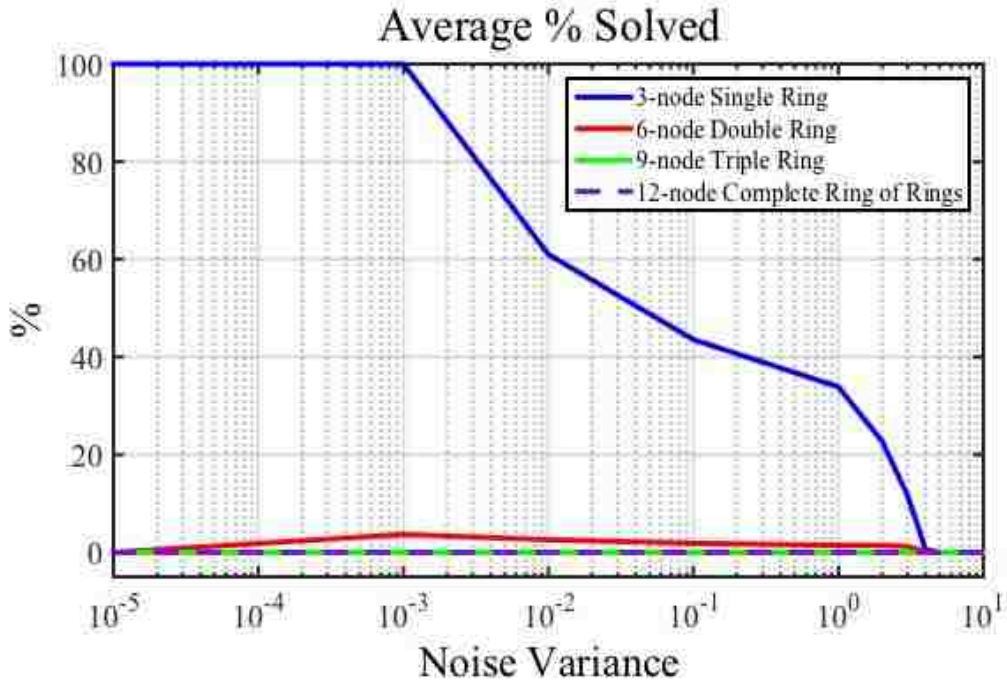


Figure 6.9: Comparison of reconstruction problems for increasing network size.

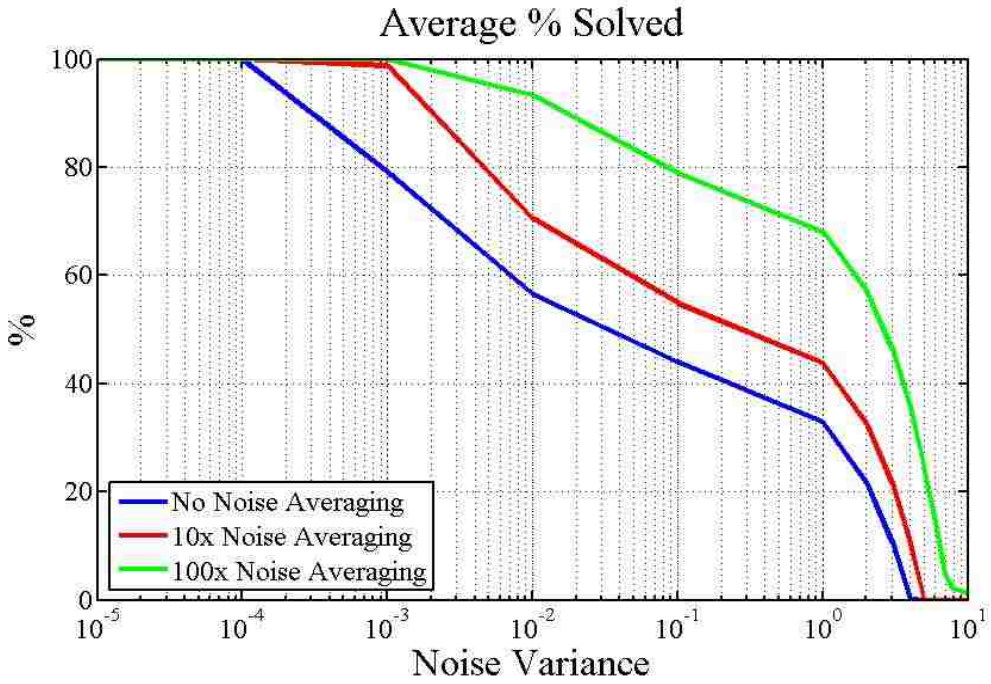


Figure 6.10: Comparison of robust network reconstruction of a single feedback loop with and without noise averaging.

Increasing Data Amount

Another way to improve the accuracy of the reconstruction process is to increase the number of data points collected during each experiment. Our conjecture is that as the size of the

network grows, the number of data points that must be collected in order to accurately reconstruct must grow exponentially, though validating this conjecture is beyond the scope of this work. Figure 6.11 shows the increase in accuracy as the number of points collected during each experiment increases for the case of the ring of rings network.

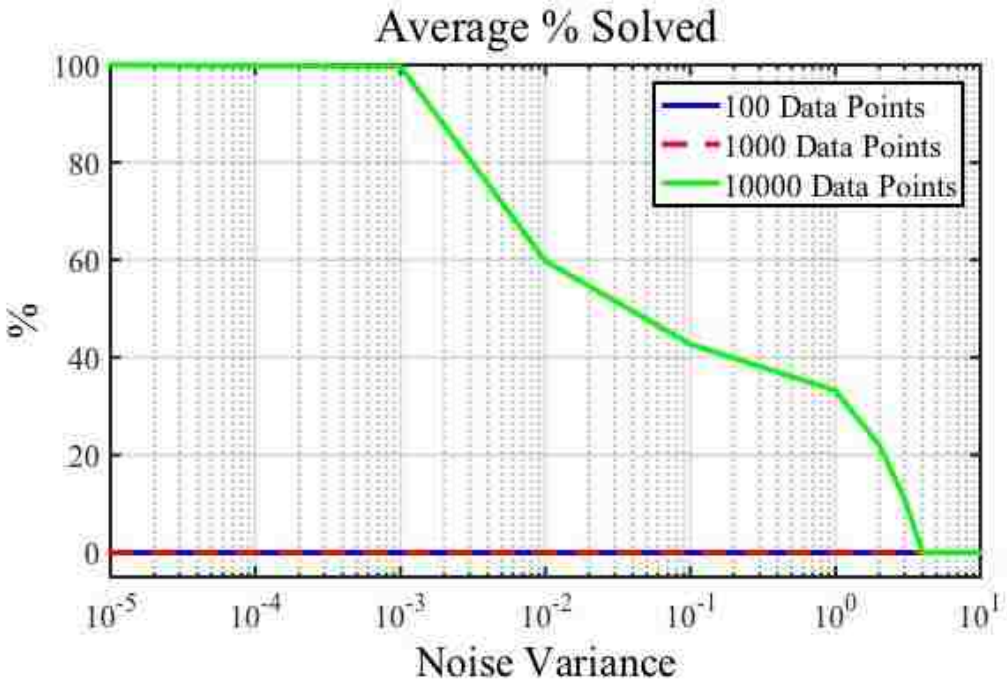


Figure 6.11: Improved reconstruction with increased data collection.

6.6 Conclusion

In this paper we extended the robust reconstruction problem beyond those networks that met the strict assumption of target specificity to include all reconstructible networks. Furthermore, we improved upon previously proposed reconstruction algorithms by further reducing the computational complexity of the reconstruction method. Then, we customized our existing model selection procedure to scale with the size of the network in order to ensure accurate reconstructions for large networks.

Chapter 7

Passive Reconstruction of the Dynamical Structure Function

(Published at ACC 2016 as "Passive Reconstruction of Non-Target-Specific Discrete-Time LTI Systems")

Abstract

Much of the existing literature on the reconstruction of a system's dynamical structure functions has focused on learning the structure of a system using experiments in which each measured state must be perturbed independently. This work develops a reconstruction procedure that does not require multiple targeted experiments, instead determining the structure of the network when inputs are drawn from a Gaussian distribution and are active simultaneously.

Although similar reconstruction procedures exist in the literature, this algorithm removes the restriction of target specificity, which states that each input must independently affect a measured state in the system. This allows for the reconstruction procedure to be applied to a larger number of networks that were previously not reconstructible because of their inherent structure. Furthermore, this is the first reconstruction procedure on the dynamical structure function to operate in the time-domain, rather than the frequency domain, in order to avoid the overhead and inaccuracies that could be introduced through transformations.

Dynamical structure functions, developed in [24], are a system representation that denote the structure and dynamics of a linear time-invariant system at a resolution consistent with the number of manifest variables. A system's dynamical structure function details the *relationship among measured states* denoted by the transfer function matrix $Q(z)$, and the *relationship between inputs and measured states*, denoted $P(z)$, where z is a variable in the frequency domain. The dynamical structure function of a system contains more information about the structure of the system than the associated transfer function, $G(z)$, which details the input-output dynamics of the system and only contains structural information about the manner in which *inputs directly affect measured states*.

Rather than reconstruct the dynamical structure function, many existing algorithms attempt to determine the state space representation of a system, (A, B, C, D), since it contains all the structural details of the system, i.e. it defines how *inputs affect internal states*, *internal states interact*, *inputs affect outputs*, and *states affect outputs*. However, determining a state space model of a system is an ill-posed problem that requires an extensive amount of a priori knowledge about the system, beyond its input-output dynamics, in order to reconstruct [38]. Note that determining the dynamical structure function from input-output data is also an ill-posed problem, but requires less a priori information to reconstruct than the system's state space realization [7], while having a stronger notion of structure than a system's transfer function.

In [73], a robust network reconstruction algorithm that determines the dynamical structure function for systems with target specificity, i.e. systems in which each input independently affects a measured state, was developed. The algorithm was improved to run in polynomial time in [28], before being extended to all reconstructible systems in [16], based on identifiability conditions from [7]. These network reconstruction techniques for dynamical structure functions, however, all were set in the frequency domain and all assumed that experiments could be performed on the system. In particular, these works perturbed each

input to *actively* probe the system and generate data informative enough to reconstruct its network structure.

Algorithms that use passive network reconstruction methods were presented in [36] and [29]; however, those were restricted to systems with target specificity, which requires each input to independently perturb a unique measured state. The novelty of the algorithm presented in this paper is that the target specificity restriction is removed which greatly increases the applicability of the reconstruction procedure. Additionally, the procedure is detailed in the time-domain in an attempt to circumvent potential inaccuracies that may be introduced through transformations on the system.

The paper begins with the derivation of the dynamical structure function and details the necessary and sufficient conditions required for network reconstruction of a system in Section 7.1. Then in Section 7.2, the dynamical structure function definition is extended from the frequency domain to the time domain. Section 7.3 details the main result, the passive reconstruction algorithm using the time domain representation of the dynamical structure function. Finally, in Section 7.4 an illustrative example of the network reconstruction algorithm applied to a system with non-diagonal $P(z)$, i.e. a system *without* target specificity, is shown.

7.1 Background

In this section we derive the dynamical structure function representation of systems and detail the necessary and sufficient informativity conditions for network reconstruction of the dynamical structure functions, assuming no measurement or process noise in the system.

7.1.1 Dynamical Structure Functions

This section gives an overview of the derivation of dynamical structure function. Consider the state space system given by:

$$\begin{aligned}\begin{bmatrix} y[k+1] \\ \psi[k+1] \end{bmatrix} &= \begin{bmatrix} A_{11}\ A_{12} \\ A_{21}\ A_{22} \end{bmatrix} \begin{bmatrix} y[k] \\ \psi[k] \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u[k] \\ y[k] \quad &= \begin{bmatrix} I\ 0 \end{bmatrix} \begin{bmatrix} y[k] \\ \psi[k] \end{bmatrix}. \end{aligned} \tag{7.1}$$

Note that $D = 0$, while $C = \begin{bmatrix} I\ 0 \end{bmatrix}$ which allows the variables to be separated into the measured states, y, and the unmeasured states, ψ.

The next step is to take the $\mathcal{Z}$-transform of the signals in (7.1). Assuming zero initial conditions, we get:

$$\begin{bmatrix} zY(z) \\ z\Psi(z) \end{bmatrix} = \begin{bmatrix} A_{11}\ A_{12} \\ A_{21}\ A_{22} \end{bmatrix} \begin{bmatrix} Y(z) \\ \Psi(z) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} U(z) \tag{7.2}$$

Solving for $\Psi(z)$, gives:

$$\Psi(z) = (zI - A_{22})^{-1} A_{21} Y(z) + (zI - A_{22})^{-1} B_2 U(z) \tag{7.3}$$

Substituting (7.3) into the first equation of (7.2) then yields

$$zY(z) = W(z)Y(z) + V(z)U(z)$$

where

$$W(z) = A_{11} + A_{12} (zI - A_{22})^{-1} A_{21}$$

and

$$V(z) = A_{12}\left(zI - A_{22}\right)^{-1} B_2 + B_1.$$

Let $D(z)$ be a matrix with the diagonal terms of $W(z)$, i.e. $D(z) = \text{diag}(W_{11}(z), W_{22}(z), ..., W_{pp}(z))$. Then,

$$\left(zI - D(z)\right) Y(z) = \left(W(z) - D(z)\right) Y(z) + V(z)U(z)$$

Note that $zI - D(z)$ is always invertible since $D(z)$ is always proper. We then have:

$$Y(z) = Q(z)Y(z) + P(z)U(z) \tag{7.4}$$

where

$$Q(z) = \left(zI - D(z)\right)^{-1}\left(W(z) - D(z)\right) \tag{7.5}$$

and

$$P(z) = \left(zI - D(z)\right)^{-1} V(z) \tag{7.6}$$

Note that since $W(z) - D(z)$ is a hollow matrix (a matrix with zeros along the diagonal), then $Q(z)$ is also a hollow matrix.

The matrix $Q(z)$ is a matrix of strictly proper transfer function from $Y_i(z)$ to $Y_j(z)$, $i \neq j$ relating each measured signal to all other measured signals. Likewise, $P(z)$ is a matrix of strictly proper transfer function from each input to each output without depending on any additional measured state $Y_i(z)$. Together, the pair $(Q(z), P(z))$ is known as the dynamical structure function of the system.

7.1.2 Necessary and Sufficient Informativity Conditions for Network Reconstruction [7]

In order to detail the necessary and sufficient conditions for network reconstruction of the dynamical structure function, we introduce the following notation. Let $A \in \mathbb{C}^{n\times m}$ and $B \in \mathbb{C}^{k\times l}$. Then:

- $\text{blckdiag}(A, B) = \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix}$,
- A_{-i} is the matrix A without its i^{th} column,
- A^T is the transpose of matrix A,
- $\mathcal{R}(A)$ is the range of A,
- $\overrightarrow{a}$ is the vector stack of the columns of A
- and $\overleftarrow{a}$ is the vector stack of the columns of A^T.

In order to determine the conditions that are necessary and sufficient for the reconstruction of the dynamical structure function from data, we construct a map of the elements of the dynamical structure function to the associated transfer function, which can be determined using system identification on input-output data. We begin by noting that the transfer function, $G(z)$, of the system in (7.1) is related to its dynamical structure, $(Q(z), P(z))$, by the following equation

$$G(z) = (I - Q(z))^{-1} P(z) \tag{7.7}$$

which can be rearranged to get:

$$\begin{bmatrix} I & G(z)^T \end{bmatrix} \begin{bmatrix} P(z)^T \\ Q(z)^T \end{bmatrix} = G(z)^T \tag{7.8}$$

Now, note that

$$\bar{A}\bar{X} = \bar{B} \iff \text{blckdiag}(\bar{\text{A}}, ..., \bar{\text{A}})\vec{\bar{x}} = \vec{\bar{b}}$$

and defining $X(z) = \begin{bmatrix} P(z)^T & Q(z)^T \end{bmatrix}$ we can then rewrite (7.8) as

$$\begin{bmatrix} I & \text{blckdiag}(G(z)^T, ..., G(z)^T) \end{bmatrix} \overrightarrow{x}(z) = \overleftarrow{g}(z). \tag{7.9}$$

Since $Q(z)$ is hollow, as noted in Section 7.1.1, we can abuse notation to redefine $\overrightarrow{x}(z)$ to remove the columns that correspond to the zero elements, reducing Equation (7.9) to the following:

$$L(z)\overrightarrow{x}(z) = \overleftarrow{g}(z). \tag{7.10}$$

where $L(z) \in \mathbb{R}^{pm \times p^2 - p + pm}$ and

$$L(z) = \left[I \text{ blckdiag}(G_{-1}(z)^T, G_{-2}(z)^T, ..., G_{-p}(z)^T) \right].$$

Identifiability conditions can then be established by determining which elements of $\overrightarrow{x}(z)$ must be known a priori in order to reduce the relationship to an injective map. To accomplish this, consider the matrix $\bar{T} \in \mathbb{R}^{p^2 - p + pm \times k}$ such that

$$\overrightarrow{x}(z) = \bar{T}h(z) \tag{7.11}$$

where $h(z)$ is an arbitrary vector of transfer functions.

Lemma 17. *Given a system characterized by the transfer function $G(z)$, its dynamical structure function $(Q(z), P(z))$ can be identified if and only if*

1. *$M(z) = L(z)\bar{T}$ is injective, i.e. $rank(M(z)) = k$, and*
2. *$\overleftarrow{g}(z) \in \mathcal{R}(M(z))$.*

Proof. The proof for Lemma 17 is given in [7]. □

7.2 Time-domain Representations of the Dynamical Structure Function

The dynamical structure function up to this point has only been defined in the frequency domain in the literature. We now extend the system representation to the time-domain, which is preferable for the development of the passive reconstruction algorithm. Previous network reconstruction techniques in the literature required system identification procedures to occur

before the reconstructive step. While this is not difficult, the added layer of complexity could lead to worse performance in the reconstruction algorithm, which is avoided by performing the reconstruction procedure directly on the data.

7.2.1 Representations

Given the dynamical structure function of the form (7.4), taking the inverse $\mathcal{Z}$-transform yields

$$y_t = Q_t * y_t + P_t * u_t \tag{7.12}$$

which we call the convolution representation of the dynamical structure function and where $*$ is the convolution operator and y_t is the output and u_t is the output at time t.

Note that (7.12) can be written in the form of matrix multiplication:

$$\bar{y}_r = \bar{Q}_r \bar{y}_r + \bar{P}_r \bar{u}_r \tag{7.13}$$

where $\bar{y}_r = \begin{bmatrix} y_1^T & y_2^T & ... & y_r^T \end{bmatrix}^T$, $\bar{u}_r = \begin{bmatrix} u_1^T & u_2^T & ... & u_r^T \end{bmatrix}^T$,

$$\bar{Q}_r = \begin{bmatrix} 0 & ... & ... & ... \\ Q_1 & \ddots & & \\ Q_2 & Q_1 & \ddots & \\ \vdots & \ddots & \ddots & \ddots \\ Q_r & & \ddots & \ddots \end{bmatrix}, \bar{P}_r = \begin{bmatrix} 0 & ... & ... & ... \\ P_1 & \ddots & & \\ P_2 & P_1 & \ddots & \\ \vdots & \ddots & \ddots & \ddots \\ P_r & & \ddots & \ddots \end{bmatrix} \tag{7.14}$$

which we call the Toeplitz representation of the dynamical structure function.

As we will see in Section 7.3.2 stable systems result in matrices $(\bar{Q}_r, \bar{P}_r)$ with entries Q_i and P_i that tend to zero as $r \to \infty$. This fact allows us to approximate $(\bar{Q}_\infty, \bar{P}_\infty)$ arbitrarily well with matrices of finite dimensions.

7.3 Main Result

Some previous reconstruction algorithms actively probe the system by constructing a sequence of experiments that independently inject an appropriate signal (e.g. a step) at each input (while holding all other inputs at zero) and measuring the entire system response [24, 73]. In situations where we have controlled access to the system inputs and can conduct such experiments, this process may be a convenient way to generate data that is very informative about the system's network structure.

However, in some situations, we may not have the ability to actively perturb each input. In these cases, the data obtained from observations of the system will not, in general, be informative enough to reconstruct network structure. Nevertheless, there has been some work where researchers have been willing to assume that the system is intrinsically perturbed by independent stochastic processes acting on each input [29, 36], although these results assume target specificity. Here, we drop the target specificity assumption, but do assume that the inputs, though not controlled, are measured and provide a persistency of excitation on the system to ensure the input-output data is rich enough for reconstruction.

7.3.1 Learning the Toeplitz Representation

The process of learning the Toeplitz representation in (7.13) is similar to learning the frequency-domain dynamical structure function using the necessary and sufficient conditions developed in [7]. First, we will extend those conditions from the frequency-domain to the time-domain, then we will discuss how the amount of data we collect will affect our estimate of r in (7.13) and thus, our current understanding of the system's structure and dynamics.

Necessary and Sufficient Conditions for Time-Domain Network Reconstruction

Given (7.14), each Q_i has $p^2 - p$ unknowns and each P_i has pm unknowns, which means in $\bar{Q}_r$ and $\bar{P}_r$ there are a total of $(p^2 - p + pm)r$ unknowns. Now, take (7.13) to be rewritten as

$$\bar{y}_r = \begin{bmatrix} \bar{Q}_r & \bar{P}_r \end{bmatrix} \begin{bmatrix} \bar{y}_r \\ \bar{u}_r \end{bmatrix}$$

Taking the transpose of both sides yields

$$\bar{y}_r^T = \begin{bmatrix} \bar{y}_r^T & \bar{u}_r^T \end{bmatrix} \begin{bmatrix} \bar{Q}_r^T \\ \bar{P}_r^T \end{bmatrix}$$

Next, we can write:

$$\begin{bmatrix} y_1^T \ ... \ y_r^T \end{bmatrix} = \begin{bmatrix} y_1^T \ ... \ y_r^T \ u_1^T \ ... \ u_r^T \end{bmatrix} \begin{bmatrix} 0 & Q_1^T & ... & Q_r^T \\ \vdots & \ddots & \ddots & \vdots \\ 0 & ... & 0 & Q_1^T \\ 0 & P_1^T & ... & P_r^T \\ \vdots & \ddots & \ddots & \vdots \\ 0 & ... & 0 & P_1^T \end{bmatrix}$$

which is equivalent to the series of equations

$$y_1^T = 0,$$
$$y_2^T = \begin{bmatrix} y_1^T & u_1^T \end{bmatrix} \begin{bmatrix} Q_1^T \\ P_1^T \end{bmatrix},$$
$$y_3^T = \begin{bmatrix} y_1^T & y_2^T & u_1^T & u_2^T \end{bmatrix} \begin{bmatrix} Q_2^T \\ Q_1^T \\ P_2^T \\ P_1^T \end{bmatrix}, etc.$$

Finally, defining

$$\hat{x} = \begin{bmatrix} Q_1^T \ ... \ Q_r^T \ P_1^T \ ... \ P_r^T \end{bmatrix}$$

allows us to write (7.13) as

$$\hat{y} = \hat{L} \overrightarrow{\hat{x}}$$

where $\hat{L} \in \mathbb{R}^{fp \times (p^2 - p + pm)r}$, f is the number of data points collected, $\hat{y} = \begin{bmatrix} y_1^T \ ... \ y_1^T \ ... \ y_f^T \ ... \ y_f^T \end{bmatrix}^T$, and

$$hatL = \begin{bmatrix} y_1^T & 0 & 0 & ... & 0 & 0 & 0 & u_1^T & 0 & 0 & ... & 0 & 0 & 0 \\ 0 & \ddots & 0 & ... & 0 & \ddots & 0 & 0 & \ddots & 0 & ... & 0 & \ddots & 0 \\ 0 & 0 & y_1^T & ... & 0 & 0 & 0 & 0 & 0 & u_1^T & ... & 0 & 0 & 0 \\ & \vdots & & & & \vdots & & & & \vdots & & & & \vdots \\ y_f^T & 0 & 0 & ... & y_1^T & 0 & 0 & u_f^T & 0 & 0 & ... & u_1^T & 0 & 0 \\ 0 & \ddots & 0 & ... & 0 & \ddots & 0 & 0 & \ddots & 0 & ... & 0 & \ddots & 0 \\ 0 & 0 & y_f^T & ... & 0 & 0 & y_1^T & 0 & 0 & u_f^T & ... & 0 & 0 & u_1^T \end{bmatrix}$$

Now, given a static matrix

$$\bar{T} = \begin{bmatrix} T_1 & T_2 \\ T_3 & T_4 \end{bmatrix},$$

representing the a priori information in the frequency domain, where T_1 is the a priori information about the system that indicates how the reduced elements of $Q(z)$ map to the original elements of $Q(z)$, with T_2 performing a similar function from the reduced elements of $Q(z)$ to the original elements of $P(z)$, and so on.

Note that $\hat{T}$ is the a priori information that maps the reduced element of each Q_i and P_i for $i = 1, ..., r$ to the unreduced elements of the appropriate matrices, and that the a priori information applies to each Q_i and P_i in the same manner, yielding

$$\hat{T} = \begin{bmatrix} T_1 & \dots & 0 & T_2 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & T_1 & 0 & \dots & T_2 \\ T_3 & \dots & 0 & T_4 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & T_3 & 0 & \dots & T_4 \end{bmatrix} \tag{7.15}$$

Note if $\bar{T}$ is not static, $\hat{T}$ is determined from the inverse $\mathcal{Z}$-transform of $\bar{T}$.

This leads us to the necessary and sufficient conditions for passive reconstruction:

Theorem 16. *Given a system characterized by the transfer function G, its dynamical structure function (Q, P) can be identified in the time domain if and only if*

1. *$\hat{M} = \hat{L}\hat{T}$ is injective, i.e. $rank(\hat{M}) = kr$,*
2. *$\hat{y} \in \mathcal{R}(\hat{M})$, and*
3. *r chosen sufficiently large.*

Proof. This follows directly from the results of Lemma 17, where $\hat{T}$ is constructed by iteratively applying $\bar{T}$ to the entries Q_i and P_i for $i = 1, ..., r$ as shown in (7.15) if $\bar{T}$ is static. If $\bar{T}$ is dynamic, $\hat{T}$ is constructed by taking the inverse $\mathcal{Z}$-transform to get T_t and then applying each T_i to Q_i and P_i for $i = 1, ..., r$. □

Incremental System Understanding

Here we discuss the notion of choosing r sufficiently large, specifically looking at the Boolean structure in terms of $\sum_{i=0}^{r} Q_i$ from reconstructed Toeplitz representation for small values of r. This will help us inform our understanding of why part 3 of Theorem 16 is a necessary condition in the time domain.

Example 17. *Consider a system $\mathcal{S}$ where the a priori information required for reconstruction is the Boolean structure of $P(z)$. In this scenario, the Boolean structure of*

$$Q(z) = \begin{bmatrix} 0\,0\,1 \\ 1\,0\,0 \\ 0\,1\,0 \end{bmatrix},$$

but this information is unknown a priori.

Using the necessary and sufficient conditions for reconstruction without r chosen sufficiently large yields the following Boolean structures

$\boldsymbol{r}$	2	5	28	120
$\boldsymbol{Q_{bool}}$	$\begin{bmatrix} 0\,1\,1 \\ 1\,0\,1 \\ 1\,1\,0 \end{bmatrix}$	$\begin{bmatrix} 0\,1\,1 \\ 1\,0\,0 \\ 1\,1\,0 \end{bmatrix}$	$\begin{bmatrix} 0\,0\,1 \\ 1\,0\,0 \\ 1\,1\,0 \end{bmatrix}$	$\begin{bmatrix} 0\,0\,1 \\ 1\,0\,0 \\ 0\,1\,0 \end{bmatrix}$

(7.16)

Note that although $r = 120$ yields the smallest r required to determine correct Boolean structure, a larger r value may be necessary in order to learn the correct dynamics of the system.

This example illustrates that the reconstruction procedure improves gradually as more data is collected from the system.

7.3.2 Learning the Convolution Representation

Given the Toeplitz representation of the dynamical structure function, the convolution representation can then be determined. First, it is necessary to determine the delays on each link. Assuming zero initial conditions, the number of time steps it takes for a link in $Q(z)$ or $P(z)$ to become non-zero is the definition of the delay on the link. This means that the first non-zero element of the $(\hat{i},\hat{j})^{th}$ entry in $Q_{\hat{k}}$ or $(\bar{i},\bar{j})^{th}$ in $P_{\bar{k}}$, for some $\hat{k},\bar{k} \in (1,...,r)$, the delays on the links are given by $w_{(\hat{i},\hat{j})} = \hat{k}$ or $w_{(\bar{i},\bar{j})} = \bar{k}$, respectively. If there are no non-zero elements, then no link exists. Given the links, we then attempt to learn the functions of the respective links in the time-domain, the form of which is derived in Theorem 17.

Theorem 17. *Given that*

$$Q(z) = (zI - D(z))^{-1}(A_{12}(zI - A_{22})^{-1})A_{21} + A_{11})$$

and

$$P(z) = (zI - D(z))^{-1}(A_{12}\left(zI - A_{22}\right)^{-1}B_2 + B_1)$$

the entries of the corresponding inverse $\mathcal{Z}$-transform will have the form

$$a_k\delta_{t,0} + \sum_{i=0}^{w_k} b_i(c_i)^t \tag{7.17}$$

where w_k is the number of delays in the corresponding link.

Proof. Consider the definition of the inverse $\mathcal{Z}$-transform which states

$$\begin{aligned} x[t] &= \mathcal{Z}^{-1}\{X(z)\} \\ &= \tfrac{1}{2\pi i}\oint_C X(z)z^{t-1}dz \\ &= \sum [\text{Residues of } X(z)z^{t-1} \text{ at the poles of } X(z)]. \end{aligned}$$

The residue of $X(z)$ at the pole z_0 is denoted

$$Res(X(z), z_0) = X(z)(z - z_0)|_{z_0}.$$

Since they are strictly proper transfer function matrices, the entries of $Q(z)$ and $P(z)$ can be written as the sum of partial fractions of the form $\frac{\alpha_i}{z-\beta_i}$. Then, defining

$$H_{j,k}(z) = Q_{j,k}(z)z^{t-1}$$

implies

$$H_{j,k}(z) = \sum_{i=0}^{w} \frac{\alpha_i z^{t-1}}{z - \beta_i}$$

The sum of the residues is then evaluated as

$$\begin{aligned} \textstyle\sum_{i=0}^{w} Res(_{j,k}(z), \beta_i) &= \alpha_i z^{t-1}|_{\beta_i} \\ &= \alpha_i \beta_i^{t-1} \\ &= \tfrac{\alpha_i}{\beta_i} \beta_i^t \end{aligned}$$

This gives us $b_i = \frac{\alpha_i}{\beta_i}$ and $c_i = \beta_i$, $\forall i \in (0, w)$. Assuming zero initial conditions then yields $a = -\sum_{i=0}^{w} b_i$.

□

Given the form in Equation (7.17) which represents the entries in Q_t and P_t, we can then determine each entry by fitting toegther the Q_i's or P_j's using a nonlinear least squares algorithm. This process can be done in MATLAB using a custom curve fit. In order to tune the fitting process, we used the following heuristic:

1. Step 1: Change the default maximum function evaluations of the model and maximum amount of iterations to $2r$ and attempt to fit the parameters.
2. Step 2: If a solution is not found, iterate through all possible maximum function evaluations and maximum amount of iterations from $v = 1, ..., 3r$.

3. Step 3: If a solution is still not found, then randomly change the start point of the fitting algorithm and repeat Steps 1 and 2 until a solution is found or a stopping condition is met, using the best solution so far.

7.3.3 Finding the Dynamical Structure Function

Once the convolution representation has been determined, take the $\mathcal{Z}$-transform of $Q(t)$ and $P(t)$ to get the dynamical structure function $Q(z)$ and $P(z)$, which is the output of the algorithm.

7.4 Numerical Example

The following example reconstructs a dynamical structure function, (Q, P), with non-diagonal P, to illustrate that this method can reconstruct systems without target specificity in the time-domain that were not previously reconstructed in this manner.

Example 18. ***(Non-Diagonal P)*** *Consider the following stable state space system:*

$$x[k+1] = \begin{bmatrix} .75 & 0 & 0 & 0 & 0 & 1.2 \\ -.1 & -.35 & 0 & 0 & 0 & 0 \\ 0 & 0 & .85 & -1 & 0 & 0 \\ 0 & -.73 & 0 & .95 & 0 & 0 \\ 0 & 0 & .43 & 0 & -.6 & 0 \\ 0 & 0 & 0 & 0 & .2 & .55 \end{bmatrix} x[k] + \begin{bmatrix} 1.4 & 0 & -1.4 \\ 0 & -.25 & 0 \\ 0 & 0 & .75 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} u[k] \tag{7.18}$$

$$y[k] = \begin{bmatrix} 1\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,0 \end{bmatrix} x[k]$$

the corresponding dynamical structure function is

$$Q(z) = \begin{bmatrix} 0 & 0 & \frac{41.28}{(4z-3)(5z+3)(20z-11)} \\ \frac{-2}{20z+7} & 0 & 0 \\ 0 & \frac{292}{(20z-17)(20z-19)} & 0 \end{bmatrix}, \quad P(z) = \begin{bmatrix} \frac{5.6}{4z-3} & 0 & \frac{-5.6}{4z-3} \\ 0 & \frac{-5}{20z+7} & 0 \\ 0 & 0 & \frac{15}{20z-17} \end{bmatrix} \tag{7.19}$$

The corresponding transfer function is full, so it does not detail any internal structure, only that all the inputs affect all the outputs, even though the system has a clear ring structure, as seen in Figure 7.1.

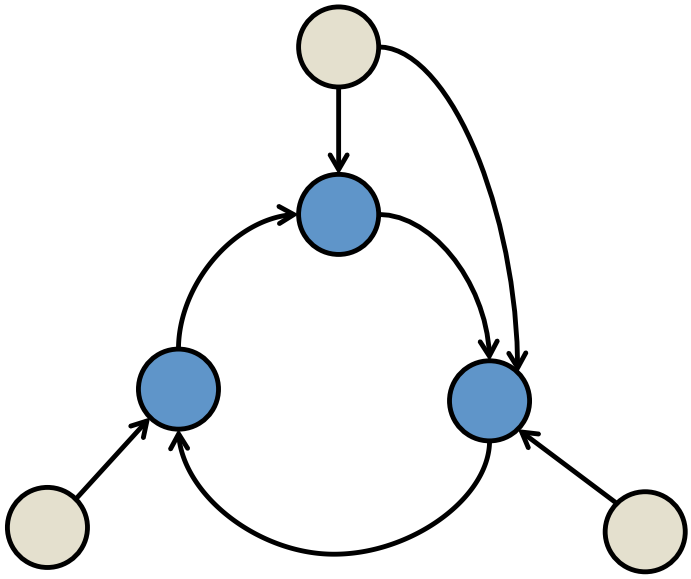


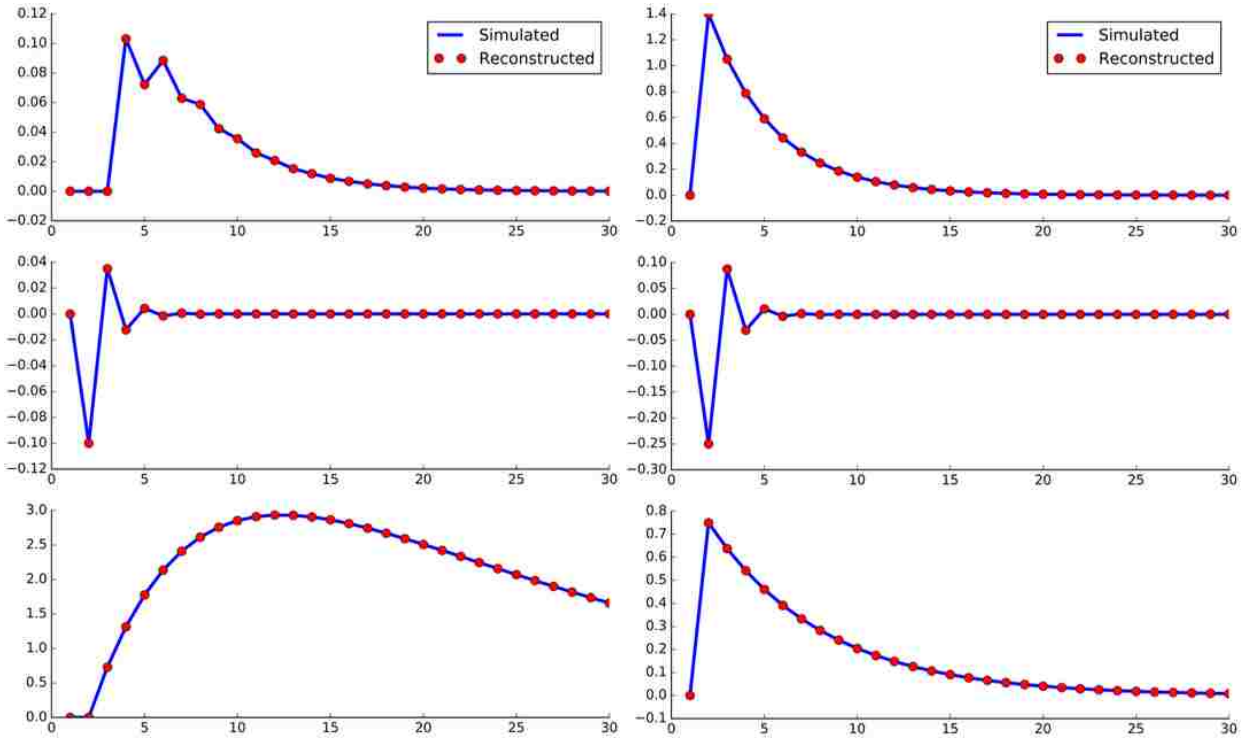
Figure 7.1: The structure of the dynamical structure function of the system in (7.18). Light nodes represent inputs and dark nodes represent measured states, with links representing causal dependencies among the manifest variables.

The convolution representation of the dynamical structure function, rounded to three decimal places, is then given by

$$\begin{aligned}
Q(t)[1,3] &= .51(.75)^t - .11(-.6)^t - .816(.55)^t + .416\delta \\
Q(t)[2,1] &= .286(-.35)^t - .286\delta \\
Q(t)[3,2] &= 7.684(.95)^t - 8.588(.85)^t + .904\delta \\
P(t)[1,1] &= 1.867(.75)^t - 1.867\delta \\
P(t)[1,3] &= -P(t)[1,1] = -1.867(.75)^t + 1.867\delta \\
P(t)[2,2] &= .714(-.35)^t - .714\delta \\
P(t)[3,3] &= .882(.85)^t - .882\delta
\end{aligned} \tag{7.20}$$

where δ represents the Kronecker delta $\delta_{(t,0)}$. This example uses the passive reconstruction method to reconstruct the dynamical structure function of the system in Equation 7.18 using simulated data and no noise. Note that $P(t)[1,3] = -P(t)[1,1]$ is part of the a priori information used to ensure that the system is reconstructible.

For this example we chose $r = 600$, noting that the longest decay (slowest dynamics) is only around 150 time steps. Typically r should be chosen so as to overestimate the actual decay of the system, as in this case. 2000 data points were collected, and the reconstruction fits are shown in Figure 7.2. The number of data points was chosen because, after being reduced for a priori information, the system has 9 unknowns in $Q(z)$ and $P(z)$, which means there are 9 unknowns in each of the (Q_i, P_i) pairs. This means that the matrix $\hat{M}$, Theorem 16, will have 5400 columns, which comes from $k = 9$ and $r = 600$. In order to achieve full column rank for $\hat{M}$ we need at least that many rows, which is why at least $f = 1800$ data points are required to reconstruct the system given that $p = 3$.



(a) Reconstructed Entries from Q (b) Reconstructed Entries from P

Figure 7.2: Each figure corresponds to an unknown entry in Q and P, with order as in (7.20). The line in each represents the simulated results of the corresponding equation. The dots represent the reconstructed values of each unknown entry.

Using the described fitting process, we can determine the entries of the convolution representation of the dynamical structure function to be:

$$\begin{aligned}
Q(t)[1,3] &= .5096(.75)^t - .1108(-.6)^t - .8158(.55)^t + .417\delta \\
Q(t)[2,1] &= .2839(-.3556)^t - .2839\delta \\
Q(t)[3,2] &= 7.684(.95)^t - 8.588(.85)^t + .904\delta \\
P(t)[1,1] &= 1.867(.75)^t - 1.867\delta \\
P(t)[1,3] &= -P(t)[1,1] = -1.867(.75)^t + 1.867\delta \\
P(t)[2,2] &= .7143(-.35)^t - .7143\delta \\
P(t)[3,3] &= .8824(.85)^t - .8824\delta
\end{aligned} \tag{7.21}$$

which is almost exactly the same as the actual convolution representation in Equation 7.20. This example shows it is possible to reconstruct a system without target specificity in the time-domain without active experiments.

7.5 Conclusion

This paper presents a passive network reconstruction algorithm, allowing for the structure and dynamics of a system to be determined directly from input-output data without the need for multiple experiments and removing the restriction of target specificity that plagues many similar reconstruction methods. Furthermore, an illustrative example showed the effectiveness of this procedure on a system without target specificity. Future work will focus on ensuring the algorithm is robust to measurement and process noise.

7.6 Appendix: Passive Reconstruction with Unmeasured Inputs

Here we demonstrate a method for passive reconstruction of systems *without* measuring inputs into the system. For notational simplicity, we revert back to the frequency domain for this explanation, but a similar result can be developed in the time domain. We start with the relationship $Y = (I - Q)^{-1}PU$, if U is unmeasured we can replace it with a variable Ψ to

represent that inputs into the system are unknown, i.e. $Y = (I - Q)^{-1}P\Psi$. This relationship can be rewritten as

$$Y = QY + P\Psi$$

Rewriting the sum as a matrix multiplication yields

$$Y = \begin{bmatrix} Q & P\Psi \end{bmatrix} \begin{bmatrix} Y \\ I \end{bmatrix}$$

Next, we take the transpose of both sides

$$Y^T = \begin{bmatrix} Y^T & I \end{bmatrix} \begin{bmatrix} Q^T \\ (P\Psi)^T \end{bmatrix}$$

Stacking vectors yields

$$\overrightarrow{y} = \begin{bmatrix} I & blckdiag(Y_{-1}^T, Y_{-2}^T, ..., Y_{-p}^T) \end{bmatrix} \overrightarrow{x} \tag{7.22}$$

where $X = \begin{bmatrix} Q^T & (P\Psi)^T \end{bmatrix}$.

In order to complete the reconstruction procedure, there are two important things to note about (7.22):

1. Solving for the unknowns, we must introduce a priori information about the system. This a priori information for unmeasured inputs is not the same as the necessary and sufficient conditions for reconstruction with measured inputs. However, it should be clear that P diagonal is a sufficient condition for reconstruction.

2. Although the structure and dynamic of Q can be learned from this process, it is not immediately evident whether or not the structure and dynamics of P can be determined, further investigation of this issue is necessary.

Chapter 8

Vulnerability Analysis for Distributed and Coordinated Destabilization Attacks

(Published at CDC 2014)

Abstract

This paper focuses on how the vulnerability of an LTI system to destabilizing attacks can be posed as its robustness to external disturbances. First, we extend existing work on single link attack models to a more generalized attack model that allows for multiple link attacks. This is done by extending the partial structure representation of dynamical structure functions to include external perturbations. Given the new model, we then discuss how to determine the vulnerability of the system for both coordinated and distributed destabilizing attacks on a system. Finally, we develop a separability result for vulnerability in feedback systems that will be useful in determining secure architectures for structured controller design.

8.1 Introduction

A variety of systems–such as power grids, water supplies, and transportation networks–can be modeled by a network of physical and cyber components. Many of these infrastructures are critical to maintaining public health and safety and it is necessary that these systems are resilient to both malicious attacks and accidental outages. Measuring the vulnerability of these systems to an attack or an unplanned outage will help to design a more robust network infrastructure.

In general, malicious attacks are usually directed towards the most sensitive parts of a system in order to increase the amount of damage they can cause with minimum effort. Attacks can take many shapes and forms, such as physical or cyber attacks against a system. Certain attacks or unplanned outages can cause cascading failures across numerous systems [62]. For example, a software bug caused what is now known as the North-east blackout of 2003. This bug blacked out several large regions of the United States and Canada for up to two days. Backup generators failed, water pressure in several cities fell, phone systems became non-operational, and major water networks were contaminated as a result [10].

The recent economic crisis has also been compared to a blackout in a power grid. The comparisons claim that a small change within the global economic network–specifically a rise in the default rates on mortgages–led to a cascading failure throughout the entire economic network much like a downed power line can lead to cascading failure within a power grid [52].

Each of these examples demonstrate how a local disturbance can cause global cascading failure throughout a networked system. Networked systems should be designed to be robust to such disturbances, especially if the disturbances could be malicious attacks. Designing systems to be robust requires models of both the system being attacked as well as the method of the attack itself. We now explore a framework of attack models which will be useful in discussing the vulnerability of LTI systems in particular.

8.1.1 Attack Models and Scenarios

The work in [64] introduces a framework by which attack scenarios may be classified. The elements of an attack may include:

- *Attack Goals:* The purpose or intent of the attack which can be stated as the impact of a successful attack on a system. This purpose may be to steal information, change the state of the system, destabilize the system, etc.
- *Attack Policy:* The mechanism by which the attacker executes the attack. This policy is characterized as a point in a three dimensional attack space, where the dimensions are defined as follows:
  - *System Knowledge:* The scope of knowledge about the system available to the attacker before the attack begins.
  - *Disclosure Resources:* The set of states that can be measured by the attacker during the attack.
  - *Disruption Resources:* The set of states that can be modified or controlled by the attacker during the attack.
- *Additional Constraints:* Any other constraints on the attack. For example, one could consider a stealthiness constraint on many attacks which further limits the amount of resources available to the attacker and the size of the attack is unobservable to the system.

Equipped with this framework, we can categorize several types of well-known attacks.

8.1.2 Deception Attacks

Deception attacks define a class of attack models that have recently received wide attention. Within the framework outlined above, deception attacks can be classified as attacks which are executed with the intent to provide false information to authorized sources [63].

8.1.3 Denial of Service Attacks

Denial of service attacks are characterized as attacks where disruption resources available to the attacker include the ability to disrupt communication between states in the system. These types of attacks may serve several purposes such as the destabilization of the system or a degradation the system's performance [69].

8.1.4 Destabilization Attacks

The primary focus of this work is on a system's vulnerability to destabilization attacks. The purpose of destabilization attacks is to create local disturbances on a network system in order to cause global cascading failures [45].

One specific attack that has recently received wide attention is Stuxnet, in large part due to the fact that, unlike many of the attacks and attack models studied in the past, the purpose of Stuxnet was not to steal, manipulate, or erase information; rather, it was intended to destroy physical systems, which it achieves by changing the input seen by the controllers on these physical systems [35]. Since Stuxnet causes global failure to these systems by making small changes to the controllers on these systems, it can be modeled as a destabilization attack.

For the purposes of this paper, destabilization attacks do not involve stealthiness constraints; however they will assert that the system knowledge as well as the disclosure resources available to the attacker are constrained to manifest or exposed states. Further, we only consider destabilization attacks on causal, linear time invariant (LTI) systems.

8.2 Background

We begin by discussing the methodology used throughout the paper for calculating the vulnerability of a link to a given attack model. The main results of this paper are developed using the dynamical structure function (DSF) representation, which is a partial structure representation of the system. We then motivate the use of the DSF representation for

vulnerability analysis of destabilizing attacks and discuss previous work already conducted in this area on single link destabilization attacks. Note that a link in the dynamical structure function represents a causal relationship between either two measured states or an input and a measured state in the state space representation of the system, possibly through hidden states.

8.2.1 Methodology for Vulnerability Analysis

Vulnerability analysis begins by deciding which system variables are potentially exposed to attack. The DSF, described in detail in Section 8.2.2, is then used to model the system in terms of these exposed variables.

An attack model specifies which exposed variables the attacker can measure and which they can affect. The system vulnerability with respect to this attack is calculated via the small gain theorem to determine how the attacker has to work to destabilize the system from its current position. The small gain theorem is applied by determining the transfer function matrix M that is in feedback with the attack Δ (see Figure 8.1).

8.2.2 Dynamical Structure Functions

The DSF representation was first developed in [24] and represents the relationship among exposed variables in a system. Given a state space representation of the form:

$$\begin{bmatrix} \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u$$
$$y = \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix}$$

where we note that $C = \begin{bmatrix} I & 0 \end{bmatrix}$ because we partition the states into the exposed states, y, and hidden states, z.

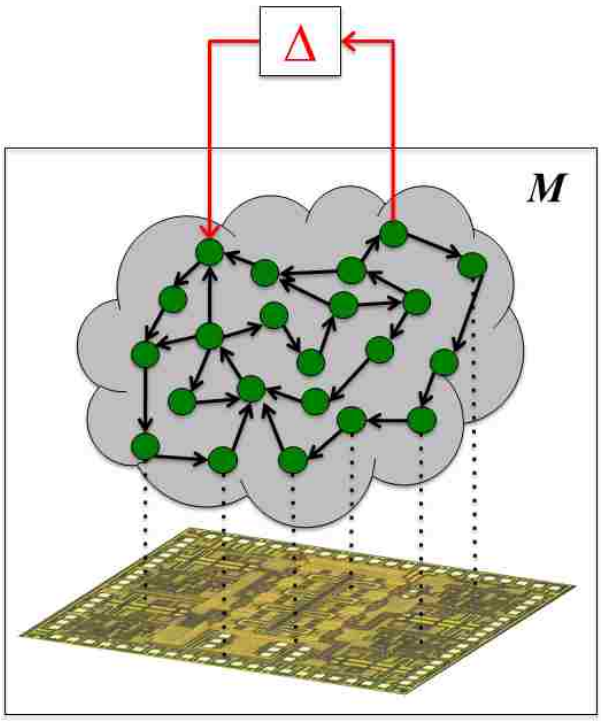


Figure 8.1: Modeling attacks as a destabilizing perturbation to a system model built around exposed variables enables standard robustness analysis to define a meaningful notion of vulnerability. The figure shows a single-link attack, resulting in a Single Input Single Output Δ block, but the methodology is similar if the attacker has access to more exposed variables, making Δ Multi Input Multi Output (in general). The structure of Δ then characterizes decentralized attacks (diagonal structure) from coordinated attacks (full-block structure).

The associated DSF (Q, P), as shown in [24], is:

$$Y = QY + PU \tag{8.1}$$

where Y and U are the Laplace transforms of y and u. Also, Q is a hollow matrix that describes how exposed states affect other exposed states, while P is a matrix that describes how inputs affect exposed states.

The relationship given in (8.1) details the structure between the exposed variables, which is the structure of the system that is most likely to be visible to an external attacker and highlights which variables an attacker can affect in the system. Therefore, the DSF is a useful representation for calculating the vulnerability of links to destabilization attacks.

8.2.3 Single Link Attack

From [45] we learn that a stable additive perturbation Δ on a link Q_{ij} or P_{ij} is able to destabilize the system if and only if the transfer function, M_{ij}, seen by Δ is nonzero. This means that the link Q_{ij} or P_{ij} is in feedback with some series of links in Q or P. Note that M_{ij} is the transfer function from ΔY_j to Y_j. From this result we can conclude that, for open-loop systems, links in the matrix P are never vulnerable to this type of an attack, so only links in Q are potentially vulnerable.

The vulnerability of a single link in Q is then defined to be the inverse of the smallest perturbation required to destabilize the system. So a large perturbation means the vulnerability is low and vice versa. By application of the small gain theorem, we note that the system will remain stable as long as $||\Delta||_\infty ||M_{ij}||_\infty < 1$.

Therefore, the smallest perturbation necessary to destabilize the system occurs when

$$||\Delta||_\infty ||M_{ij}||_\infty = 1$$

which means

$$||\Delta||_\infty = \frac{1}{||M_{ij}||_\infty}$$

Thus, the vulnerability of a link Q_{ij} is

$$v_{ij} = ||M_{ij}||_\infty$$

The overall vulnerability, V, of the entire system to single link attacks can then be defined as the largest vulnerability of any single link, looking across all links in the system; i.e.

$$V = \max_{Q_{ij} \neq 0 \in Q} v_{ij}$$

8.3 State Space Attack Model

The analysis of the vulnerability of a single link to a destabilizing attack can then be extended to a more general model incorporating simultaneous attacks on multiple links. We begin by formulating this model using the state space representation of LTI systems.

We model an attack on a network system as an external disturbance $F\psi$ to the system to get:

$$\begin{aligned} \dot{x} &= Ax + Bu + F\psi \\ y &= \begin{bmatrix} I & 0 \end{bmatrix} x \end{aligned} \tag{8.2}$$

As noted in Section 8.1, the purpose of the attack is to destabilize the system using a well-defined set of disclosure and disruption resources. Restricting ourselves to the class of stable systems we can see that a bounded input will create a bounded output, so the system can not be destabilized by a stable external disturbance, so we restrict ourselves to the case where $\psi = x$.

With $\psi = x$ an attacker is allowed to artificially create a link connecting state x_i to state x_j; moreover, if $F = B$ we can artificially create a link from a state x_i to an input u_j. Allowing the attacker to create links that have feedback with existing links has negative implications for the overall vulnerability of the system. This is discussed in more detail in Section 8.4.1.

For many attack models, a reasonable restriction may be that the attacker is only allowed to use the existing system infrastructure. Therefore, we can restrict this attack model such that if $a_{ij} = 0$, then $\delta_{ij} = 0$. We can restrict the attacker further to only allow attacks on exposed states. For this restricted class of attacks, a more useful representation of a system is the DSF. Therefore, our next step is to formulate a generalized attack model in the DSF framework before adding these restrictions.

8.4 A Generalized Attack Model using the Dynamical Structure Function Representation

In order to define a generalized attack model in the DSF domain, we first need to define the DSF of a system when external disturbances are present. We rewrite (8.2) to be a state space system of the form

$$\begin{aligned} \begin{bmatrix} \dot{y} \\ \dot{z} \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u + \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \psi \\ y &= \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} \end{aligned} \tag{8.3}$$

Taking the Laplace transform of (8.3) yields

$$\begin{bmatrix} sY \\ sZ \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} Y \\ Z \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} U + \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \Psi \tag{8.4}$$

where Z and Ψ are the Laplace transforms of z and ψ, respectively.

Solving for Z in the second equation of (8.4) gives us

$$Z = (sI - A_{22})^{-1} A_{21} Y + (sI - A_{22})^{-1} B_2 U + (sI - A_{22})^{-1} F_2 \Psi \tag{8.5}$$

Plugging (8.5) into the first equation in (8.4) then gives us

$$sY = WY + VU + N\Psi \tag{8.6}$$

with $W = A_{11} + A_{12}(sI - A_{22})^{-1} A_{21}$, $V = B_1 + A_{12}(sI - A_{22})^{-1} B_2$, $N = F_1 + A_{12}(sI - A_{22})^{-1} F_2$

Finally, defining $D = diag(W_{11}, ..., W_{pp})$ subtracting DY from both sides of (8.6) we get

$$Y = QY + PU + \Delta\Psi \tag{8.7}$$

where $Q = (sI - D)^{-1}(W - D)$, $P = (sI - D)^{-1}V$, and $\Delta = (sI - D)^{-1}N$.

Equation (8.7) is then a generalized attack model in the DSF domain.

8.4.1 Calculating Vulnerability from a Generalized Attack Model

The class of attacks we focus on in this paper is the class of destabilizing attacks, which in turn can be split into two classes of attacks: the first in which the attacker can create communication links in a network and the second in which the attacker must use the existing communication links.

If we allow an attacker to create links in a system then it is possible for an attacker to create a feedback loop in any system. And as shown in Section 8.4.2, this feedback loop will create a vulnerability within the system. Therefore, under this assumption, no completely secure architecture can exist. For this reason and from this point forward, we only consider attack models that use the existing communication structure to conduct an attack. This is not an unreasonable assumption since creating new links within a system may be a difficult or expensive task for an attacker.

Vulnerability of a Single Link Attack

Starting with (8.7) and solving for Y in terms of U and Ψ we get

$$Y = (I - Q)^{-1}PU + (I - Q)^{-1}\Delta\Psi \tag{8.8}$$

where the input-output relationship is given by $G = (I - Q)^{-1}P$ and the transfer function describing how Ψ affects the exposed states, Y, is $(I - Q)^{-1}\Delta$. Given that we are only considering stable systems, we know that no bounded input can destabilize the system. Therefore, we consider the case when $\Psi = Y$, which means that an attacker is using some combination of additive perturbations on existing links to destabilize the system.

The transfer function seen by a perturbation ΔY is then given in (8.8) as $(I-Q)^{-1}$. In particular, if we want to determine the vulnerability of a single link attack on a link Q_{ij}, we know this can be modeled as $\Delta_{ij} = (sI - D_{ii})^{-1}N_{ij}$ with the rest of the entries in Δ equal to zero. Then, the transfer function seen by the perturbation on the link Q_{ij} is found from

$$\begin{bmatrix} Y_1 \\ \vdots \\ Y_{j-1} \\ Y_j \\ Y_{j+1} \\ \vdots \\ Y_p \end{bmatrix} = H \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \Delta_{ij}Y_j \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{8.9}$$

where $H = (I-Q)^{-1}$. From (8.9), we can see that $Y_j = H_{ji}\Delta_{ij}Y_j$ since $\Delta_{ij}Y_j$ is in the i^{th} row of the vector given in (8.9). Therefore,

$$v_{ij} = ||H_{ji}||_\infty$$

which means that

$$V = \max_{Q_{ij}\neq 0 \in Q} ||H_{ji}||_\infty \tag{8.10}$$

Vulnerability of a Multiple Link Distributed Attack

We now consider an attack in which multiple attackers are simultaneously performing unique single link attacks in the system and are not sharing information. This is modeled by the concatenation of several single link attacks on the system and by application of the small gain theorem, the vulnerability, $v_{ij,\ldots,kl}$ of this type of an attack is the structured singular

value, $\mu_{ij,\ldots,kl}$, of the matrix

$$R_{ij,\ldots,kl} = \begin{bmatrix} H_{ji} & 0 & \ldots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & H_{lk} \end{bmatrix} \tag{8.11}$$

That is,

$$v_{ij,\ldots,kl} = \mu(R_{ij,\ldots,kl}, \Pi)$$

The overall vulnerability of the system of a distributed attack is

$$V = \max_{R_{links} \in \mathscr{R}} \mu_{links}$$

where $\mathscr{R}$ is the set of matrices of the form (8.11) over the set of all possible combinations of links, $\mathscr{L}$, and μ_{links} is the structured singular value of R_{links}.

Vulnerability of a Multiple Link Co-ordinated Attack

A multiple link coordinated attack is a generalizaation of a single link attack and is similar to a distributed attack, except that it models either communication between multiple attackers or a single attacker targeting multiple links. The transfer function seen by a perturbation on multiple links when allowing for communication in the attack is then given by

$$T_{ij,\ldots,kl} = \begin{bmatrix} H_{ji} & \ldots & H_{li} \\ \vdots & \ddots & \vdots \\ H_{jk} & \ldots & H_{lk} \end{bmatrix} \tag{8.12}$$

In this case, we find the vulnerability to be

$$v_{ij,\ldots,kl} = ||T_{ij,\ldots,kl}||_\infty$$

Thus the overall vulnerability of the system of a co-ordinated attack is then given by

$$V = \max_{links \in \mathscr{L}} ||T_{links}||_\infty$$

8.4.2 Reducing Vulnerability in Open-Loop Systems

Since the vulnerability of any given link in a system is the transfer function seen by a perturbation on that link the vulnerability of the system is nonzero if and only if feedback exists within the system. Therefore, one completely secure architecture is one in which no links in Q exist.

Note that since $G = (I - Q)^{-1}P$, when $Q = 0$, then $P = G$. Since links in P are never in feedback for open-loop systems in which attackers cannot create links, they are never vulnerable. Thus the overall vulnerability of a system with $Q = 0$ is $V = 0$, meaning there does not exist a finite additive perturbation on a link in the system that can destabilize the system under the assumption that the attacker can only use the existing communication network of the system [45].

However, if an attacker is allowed to create arbitrary links within the system, any system with at least one link will be vulnerable since the attacker can create a link in feedback with the existing link.

8.5 Feedback Systems

Although completely secure systems can be created by removing all feedback from the system, in many situations, feedback is a necessary component of the design of the system and cannot

be removed. For example, an unstable system can be stabilized through a controller connected in feedback.

Throughout this section, we will refer to feedback systems as an autonomous system containing a plant with behavior G such that $Y = GU$ and a controller with behavior K such that $U = KY$, and where $G \neq K^{-1}$. Note that the plant and the controller may each be stable or unstable; however, we only consider the cases where the combined feedback systems are BIBO stable.

8.5.1 Feedback Systems as a Dynamical Structure Functions

For a feedback system, we consider an attack Ψ which may attack either the plant or the controller. Again, we are only considering stable systems and since no bounded input can destabilize a stable system, we only consider the case where $\Psi = \begin{bmatrix} Y & U \end{bmatrix}^T$.

Let the structure of G be given by (P_g, Q_g) and consider a general attack $\Delta_g \Psi = \begin{bmatrix} \Delta_{11} & \Delta_{12} \end{bmatrix} \Psi$ on the plant. Then, from (8.7), we get

$$Y = Q_g Y + P_g U + \begin{bmatrix} \Delta_{11} & \Delta_{12} \end{bmatrix} \Psi.$$

Similarly, let the structure of K be given by (P_k, Q_k) and consider a general attack $\Delta_k \Psi = \begin{bmatrix} \Delta_{21} & \Delta_{22} \end{bmatrix} \Psi$ on K. Then

$$U = Q_k U + P_k Y + \begin{bmatrix} \Delta_{21} & \Delta_{22} \end{bmatrix} \Psi.$$

Combining these equations, we get

$$\begin{bmatrix} U \\ Y \end{bmatrix} = \begin{bmatrix} Q_k & P_k \\ P_g & Q_g \end{bmatrix} \begin{bmatrix} U \\ Y \end{bmatrix} + \begin{bmatrix} \Delta_{11} & \Delta_{12} \\ \Delta_{21} & \Delta_{22} \end{bmatrix} \Psi$$
$$\triangleq Q_c \begin{bmatrix} U \\ Y \end{bmatrix} + \Delta_c \Psi \tag{8.13}$$

The combined system in (8.13) is a DSF; therefore we can perform a vulnerability analysis on Q_c in the same manner as before. Note that the links in P_k and P_g are now in the Q of the combined system; therefore, it is possible for these links to have a nonzero vulnerability. We can also define Δ_{11} as an attack on Q_k, Δ_{12} as an attack on P_k, and so forth.

8.5.2 Separability in Feedback Systems

One property of vulnerability within feedback systems is the notion of separability. We define separability as follows:

Definition 24. *Consider all potential multiple link coordinated attacks on a controller (of which single link attacks are a subset). The vulnerability analysis on attacks in the controller are considered* separable *from the plant if the computation of the vulnerability of any and all attacks in the controller depends only on the behavior G of the plant and not on the structure (P_g, Q_g) of the plant.*

Note that we can make a similar definition for the separability of vulnerability of the links in the plant from the controller. Equipped with this definition, we can then show the conditions under which multiple link coordinated attacks allow for separability.

Theorem 18. *The vulnerability analysis of the controller resulting from any arbitrary multiple link coordinated attack is separable from the plant if and only if the attack does not include*

an attack on the plant. In other words, the vulnerability analysis of the controller is separable from the plant if and only if Δ_{21} *and* Δ_{22} *are zero.*

Proof. Let Q_k and Δ_{11} be size $p \times p$. Also let P_k, K, and Δ_{12} be size $p \times m$. Consider also $H_c = (I - Q_c)^{-1} = \begin{bmatrix} \Gamma_1 \, \Gamma_2 \\ \Gamma_3 \, \Gamma_4 \end{bmatrix}$. Then we have

$$\begin{bmatrix} (I - Q_k) & -P_k \\ -P_g & (I - Q_g) \end{bmatrix} \begin{bmatrix} \Gamma_1 \, \Gamma_2 \\ \Gamma_3 \, \Gamma_4 \end{bmatrix} = \begin{bmatrix} I \, 0 \\ 0 \, I \end{bmatrix}.$$

Thus

$$\begin{aligned}
(I - Q_g)\Gamma_3 - P_g\Gamma_1 &= 0, \\
\Gamma_3 &= (I - Q_g)^{-1} P_g \Gamma_1 = G\Gamma_1, \\
(I - Q_k)\Gamma_1 - P_k\Gamma_3 &= I, \\
(I - Q_k)\Gamma_1 - P_k G\Gamma_1 &= I, \\
\Gamma_1 &= (I - Q_k - P_k G)^{-1} \\
&= ((I - Q_k) - (I - Q_k)KG)^{-1} \\
&= ((I - Q_k)(I - KG))^{-1} \\
&= (I - KG)^{-1}(I - Q_k)^{-1}, \\
\Gamma_3 &= G\Gamma_1 = G(I - KG)^{-1}(I - Q_k)^{-1}
\end{aligned}$$

By similar logic, we can find Γ_2 and Γ_4, resulting in

$$H_c = (I - Q_c)^{-1} = \begin{bmatrix} (I - KG)^{-1}(I - Q_k)^{-1} & K(I - GK)^{-1}(I - Q_g)^{-1} \\ G(I - KG)^{-1}(I - Q_k)^{-1} & (I - GK)^{-1}(I - Q_g)^{-1} \end{bmatrix} \tag{8.14}$$

It can be shown that Γ_1 is size $p \times p$ and Γ_3 is size $m \times p$.

From (8.12), we have that the vulnerability of this attack will be $V = \max_{links \in \mathscr{L}} \|T_{links}\|_\infty$, where each entry with label $\{H_c\}_{ji}$ corresponds to a nonzero attack Δ_{ij}.

Assume that an arbitrary attack occurs only on the controller. Then we have that every entry in Δ_{21} and Δ_{22} is zero and all nonzero entries $\delta_{ij} \in \Delta$ will exist where $i \leq p$. Thus all entries with label $\{H\}_{ji}$ in T_{links} are taken from the first p columns of H_c. Since the width of Γ_1 and Γ_3 is p, we have that V is dependent only on Γ_1 and Γ_3. And since Γ_1 and Γ_3 are dependent only on G, K, and Q_k, we have that the vulnerability of this attack is independent from the structure of the plant.

Now assume that the computation of vulnerability depends on Q_g. Then, from (8.14), at least one of the entries of T_{links} must be in Γ_2 or Γ_4. Also, because $(I - Q_g)^{-1}$ is full rank, at least one entry in each of Γ_2 and Γ_4 must be nonzero. Without loss of generality, assume that this entry is $\{\Gamma_2\}_{11}$. Since Γ_1 has width p, $\{\Gamma_2\}_{11} = \{H_c\}_{1,(p+1)}$. The corresponding attack that would lead to this requires that $\Delta_{(p+1),1} \neq 0$. And since Δ_{11} has height p, $\Delta_{(p+1),1} = \{\Delta_{21}\} \neq 0$; therefore, there exists a nonzero attack on the plant that depends on Q_g.

□

Corollary 4. *The vulnerability analysis of the plant resulting from any arbitrary multiple link coordinated attack is separable from the controller if and only if the attack does not include an attack on the controller.*

Proof. This follows by switching the roles of the plant and the controller in Theorem 18. □

Recall that single link attacks are a special case of multiple link coordinated attacks; therefore, when considering single link attacks on the controller, by Theorem 18, the vulnerability analysis on links in the controller are guaranteed to be separable (and a similar statement can be said about the plant).

8.5.3 Design Questions for Minimizing Vulnerability in Feedback Systems

As with open-loop systems, we consider the problem of minimizing the vulnerability of feedback systems. In particular, we hold the input-output behavior G and K of our plant and controller constant and then seek to change the structures (P_g, Q_g) and (P_k, Q_k) of the plant and the controller in order to minimize the vulnerability of the combined feedback system. To date, this problem remains unsolved. However, separability may provide important insights into the solution for this problem.

For simplicity through this discussion, we consider only single link attacks; therefore, separability is guaranteed in all cases. Note, however, that the implications analyzed within this discussion will extend to multiple link attacks so long as each attack is restricted to only attacking the plant or only attacking the controller.

The vulnerability of the combined system is then defined as the most vulnerable link within the combined system. We also define the vulnerability of the controller as the vulnerability of the most vulnerable link within the controller after it is connected in feedback to the plant (and we define the vulnerability of the plant in a similar manner).

Given these definitions, separability shows the following:

- The structure (P_k, Q_k) of the controller can be freely changed without inadvertently increasing the vulnerability of the plant.
- The structure (P_k, Q_k) of the controller cannot be changed in order to reduce the vulnerability that exists within the plant.

Consider now a sub-problem to the problem above where we hold (P_g, Q_g) constant. Separability shows that, when minimizing vulnerability, the following scenarios may occur:

- We find a structure (P_k, Q_k) of the controller such that the vulnerability of the controller is the vulnerability of the plant. Then there is nothing more we can do to minimize the vulnerability of the combined system; the vulnerability is locked at the vulnerability of

the plant. In this case, a better problem may be to minimize the vulnerability of the controller rather than minimizing the vulnerability of the combined system.

- We find that there exists no structure (P_k, Q_k) of the controller such that the vulnerability of the controller is less than the vulnerability of the plant. Then the vulnerability of the least vulnerable controller will be the minimum vulnerability of the combined system.

8.6 Conclusion

In conclusion, we have developed a generalized attack model in the dynamical structure function domain that can handle a multitude of attack vectors including accidental failures in a system. Using this model we showed how to calculate the vulnerability of a system for destabilizing attacks on a networked system. In future work we plan to determine the vulnerability of systems in the face of other attack models, including state hijacking, which looks at how bounded inputs can change state trajectory without destabilizing the system, and inference threats, where an attacker does not affect the system, but can listen to traffic along communication links.

Furthermore, we also noted that to reduce the vulnerability of a system in the face of destabilization it is necessary to remove all feedback from the system, thus the only secure architecture across all models for attacks that use existing communication links occurs when $Q = 0$. Finally, we noted that for cases when removing feedback is impossible, there is a principle of separability in which the computation of vulnerability for attacks on the controller is separable from the structure of a plant when the plant or controller are not attacked simultaneously. This result will hopefully lead to a method for minimizing vulnerability in systems where feedback is essential, which will be explored in future work.

Chapter 9

Conclusions and Future Work

9.1 Contributions

The primary goal of this thesis was to understand the process of network inference for linear time invariant systems. We established a theory of structure for dynamical systems, detailed relationships between different notions of structure, and developed applications for network reconstruction.

9.1.1 Theory

The theoretical contributions of this work were as follows:

1. Expanded the definition of the dynamical structure function to the class of all linear time invariant state space models.
2. Generalized necessary and sufficient conditions for identifiability of the dynamical structure function to any linear time invariant system.
3. Established a methodology for comparing network semantics of various system representations which demonstrated that the subsystem structure and signal structure are distinct representations.
4. Developed necessary and sufficient conditions for the identifiability of a class of structured linear fractional transformation and showed that the cost of reconstruction of the dynamical structure function is always less than or equal to the cost of reconstruction of the structured linear fractional transformation.

9.1.2 Applications

The applications discussed in this work were as follows:

1. Detailed a robust polynomial reconstruction algorithm for all systems, both target specific and non-target specific, with lower computational complexity than current algorithms.
2. Developed a passive, time-domain, reconstruction algorithm of the dynamical structure function, previous algorithms were all developed in the frequency domain.
3. Used the dynamical structure function to represent the attack surface of a system and extended previous vulnerability results from single link attacks to multiple link attacks, both coordinated and uncoordinated.

9.2 Future Work

Although much work has been done in the literature to study linear systems and the relationship between different system representations, the development of the signal structure and its associated dynamical structure function has created many new avenues for future research.

9.2.1 Definitions of System Representations

Although we extended the definition of the dynamical structure function, there is still other representations of systems (linear and nonlinear) that require more attention.

Definition of the Structured Linear Fractional Transformation

The definition of the structured linear fractional transformation should be extended to

$$N = \begin{bmatrix} J & H \\ L & K \end{bmatrix} \text{ and } S(s) = \begin{bmatrix} S_1(s) & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & S_q(s) \end{bmatrix} \tag{9.1}$$

in order to model the class of all interconnected proper transfer functions. Moreover, removing some of the restrictions imposed in Section 5.2.3, including cases where L are not diagonal, such as the system given in Figure 9.1 will be an important step.

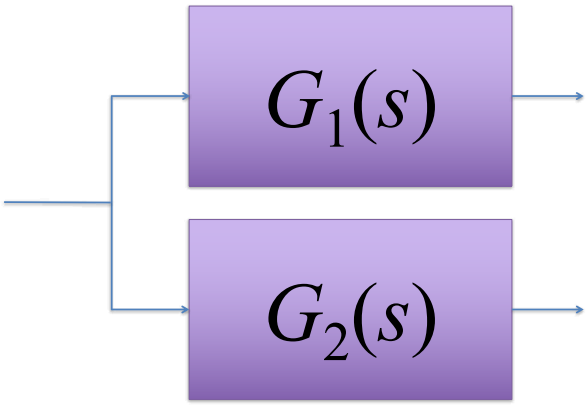


Figure 9.1: A subsystem structure with a single input affecting multiple subsystems, the associated structured linear fractional transformation will have L be a non-diagonal matrix.

Definitions of Nonlinear System Representations

This thesis detailed relationships of representations of linear systems, so a natural extension would be to determine the definitions and relationships between the corresponding nonlinear system representations associated with each of the four graphical structures outlined in this work.

9.2.2 Network Semantics

In preparation for further developments in the area of network semantics and since the state space model does not uniquely define a structured linear fractional transformation we need a representation of higher structural informativity than a state space model. One potential representation is the generalized state space equations [70] which take the form

$$\begin{aligned} \dot{x} &= Ax + \hat{A}w + Bu \\ w &= \bar{A}x + \tilde{A}w + \bar{B}u \\ y &= Cx + \bar{C}w + Du \end{aligned} \tag{9.2}$$

where $w \in \mathbb{R}^l$ is known as the auxiliary variables and l is known as the intricacy. In order to determine the semantics of system representations in terms of the generalized state space equations, a framework will be needed for describing the semantics of the representation, a nontrivial task.

Semantics of the Dynamical Structure Function

Two areas of the current network semantics results that demand more reflection are:

1. The necessary conditions on a transformation that maintains the intermediate W and V representation of a dynamical structure function. Even though the conditions are based on the state space representation rather than the generalized state space equations, the results are likely to inform the new semantics of the generalized equations.
2. Given the necessary conditions for maintaining W and V, in order to show necessary conditions for maintaining the dynamical structure function, we need to determine conditions for the case when $(W(s), V(s)) \neq (\hat{W}(s), \hat{V}(s))$. If

$$(Q(s), P(s)) = (\hat{Q}(s), \hat{P}(s)) \tag{9.3}$$

$$((sI - D_W(s))^{-1}(W(s) - D_W(s)), (sI - D_W(s))^{-1}V(s)) = \tag{9.4}$$

$$((sI - \hat{D}_{\hat{W}}(s))^{-1}(\hat{W}(s) - \hat{D}_{\hat{W}}(s)), (sI - \hat{D}_{\hat{W}}(s))^{-1}\hat{V}(s))$$

Equation 9.4 can be rewritten as the set of equations:

$$\begin{aligned} \hat{W}(s) &= (sI - \hat{D}_{\hat{W}}(s))(sI - D_W(s))^{-1}(W(s) - D_W(s)) + \hat{D}_{\hat{W}}(s) \\ \hat{V}(s) &= (sI - \hat{D}_{\hat{W}}(s))(sI - D_W(s))^{-1}V(s) \end{aligned} \tag{9.5}$$

This means that the dynamical structure function is maintained by any transformation that meets the criteria in (9.5), but has $(W(s), V(s)) \neq (\hat{W}(s), \hat{V}(s))$. One of the issues for determining the conditions on the transformation is that it is possible to maintain $Q(s)$ and $P(s)$ with $W(s) \neq \hat{W}(s)$ when no transformation exists due to the orders of the associated realizations being different; This idea is demonstrated in Example 19.

Example 19. *Consider a state space model with:*

$$\begin{aligned} A_{11} &= \begin{bmatrix} -1 & 2 \\ 2 & -1 \end{bmatrix}, A_{12} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \\ A_{21} &= \begin{bmatrix} 2 & 2 \end{bmatrix}, \qquad A_{22} = [-1] \end{aligned} \tag{9.6}$$

which means that

$$A = \begin{bmatrix} -1 & 2 & 1 \\ 2 & -1 & 2 \\ 2 & 2 & -1 \end{bmatrix} \text{ and } C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{9.7}$$

which means the system is 3^{rd} order. One of the corresponding intermediate matrices is given by

$$W(s) = \begin{bmatrix} \frac{-s+1}{s+1} & \frac{2s+4}{s+1} \\ \frac{2s+6}{s+1} & \frac{-s+3}{s+1} \end{bmatrix} \tag{9.8}$$

which means the associated $Q(s)$ matrix of the dynamical structure function is then

$$Q(s) = \begin{bmatrix} 0 & \frac{2s^2+6s+4}{s^3+3s^2+s-1} \\ \frac{2s^2+8s+6}{s^3+3s^2-s-3} & 0 \end{bmatrix} \tag{9.9}$$

Now, we want to find a state space model with the same dynamical structure function, i.e. $Q(s) = \hat{Q}(s)$, but a different intermediate matrix $\hat{W}(s) \neq W(s)$. So we define:

$$\hat{D}_{\hat{W}}(s) = \begin{bmatrix} \frac{1}{s+1} & 0 \\ 0 & \frac{1}{s+2} \end{bmatrix}. \tag{9.10}$$

Now, given (9.5) and (9.10) we can get

$$\hat{W}(s) = \begin{bmatrix} \frac{1}{s+1} & \frac{2s^4+8s^3+8s^2-2s-4}{s^4+4s^3+4s^2-1} \\ \frac{2s^4+12s^3+20s^2+4s-6}{s^4+5s^3+5s^2-5s-6} & \frac{1}{s+2} \end{bmatrix} \tag{9.11}$$

whose associated dynamical structure function is $\hat{Q}(s) = Q(s)$. *Since* $\hat{W}(s) = \hat{A}_{11} + \hat{A}_{12}(sI - \hat{A}_{22})^{-1}\hat{A}_{21}$ *we can determine the associated realization:*

$$\hat{A}_{11} = \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix} \qquad \hat{A}_{12} = \begin{bmatrix} -1.2982 & 0.2316 & 0 & 0.1486 & 0.0074 \\ 0 & 0 & 2.2857 & -0.05 & 1 \end{bmatrix}$$

$$\hat{A}_{21} = \begin{bmatrix} -0.1213 & 0.2662 \\ 3.638 & 3.6509 \\ 0.5833 & 0 \\ -0.0333 & -3.4053 \\ 0.665 & 0.8297 \end{bmatrix} \quad \hat{A}_{22} = \begin{bmatrix} 0.4126 & 0.0471 & 0 & 0 & 0 \\ 0.0471 & -0.9984 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -2.4132 & -0.0207 \\ 0 & 0 & 0 & -0.0207 & -2.001 \end{bmatrix} \tag{9.12}$$

which means that

$$\hat{A} = \begin{bmatrix} 0 & 2 & -1.2982 & 0.2316 & 0 & 0.1486 & 0.0074 \\ 2 & 0 & 0 & 0 & 2.2857 & -0.05 & 1 \\ -0.1213 & 0.2662 & 0.4126 & 0.0471 & 0 & 0 & 0 \\ 3.638 & 3.6509 & 0.0471 & -0.9984 & 0 & 0 & 0 \\ 0.5833 & 0 & 0 & 0 & 1 & 0 & 0 \\ -0.0333 & -3.4053 & 0 & 0 & 0 & -2.4132 & -0.0207 \\ 0.665 & 0.8297 & 0 & 0 & 0 & -0.0207 & -2.001 \end{bmatrix} \tag{9.13}$$

which is 7^{th} *order.*

Therefore:

1. *In some cases, no transformation exists that can maintain the dynamical structure function when* $W(s) \neq \bar{W}(s)$ *because the order of the original system is not preserved, and*

2. *One possible way of finding a necessary condition for the set of transformations that maintains the dynamical structure function is to determine the properties of the set of $\hat{D}_{\hat{W}}(s)$ matrices from (9.5) that do not change the order of the associated state space model. Then, given two different state space models of the same order, we can determine the properties of T that relate the two and gain insight into a general set of properties for necessary conditions on transformations that maintain the dynamical structure function.*

9.2.3 Applications

Finally, there is plenty of room for expansion of the structural applications discussed in this thesis. The passive reconstruction algorithm requires an extension for robust reconstruction in the presence of noise. While the polynomial-time robust reconstruction algorithm may contain some ideas that can be applied, the curve fitting algorithm struggles with noisy data and will require more careful thought. In terms of the vulnerability application, the problem of feedback vulnerability is still an open problem. Feedback vulnerability refers to a methodology for reducing the overall vulnerability of a dynamical structure function in cases when feedback is inherent in the system.

References

[1] Eiffel Tower. http://www.history.com/topics/eiffel-tower. Accessed: 2016-09-26.

[2] Guernica (Picasso). https://en.wikipedia.org/wiki/Guernica_(Picasso). Accessed: 2016-10-01.

[3] System Builder Marathon, Dec. 2009: $1,300 Enthusiast PC. http://www.tomshardware.com/reviews/build-own-pc,2503-5.html. Accessed: 2016-09-26.

[4] Interacting Proteins. https://vpr-norman.ou.edu/interacting-proteins. Accessed: 2016-09-26.

[5] Semantics. https://en.oxforddictionaries.com/definition/semantics. Accessed: 2016-10-03.

[6] Structure. http://www.merriam-webster.com/dictionary/structure. Accessed: 2016-09-26.

[7] Julius Adebayo, Taylor Southwick, Vasu Chetty, Enoch Yeung, Ye Yuan, Jorge Gonçalves, Julianne Grose, John Prince, Guy-Bart Stan, and Sean Warnick. Dynamical Structure Function Identifiability Conditions Enabling Signal Structure Reconstruction. *Conference on Decision and Control*, pages 4635–4641, 2012.

[8] Alborz Alavian and Michael Rotkowitz. Stabilizing Decentralized Systems with Arbitrary Information Structure. *Conference on Decision and Control*, pages 4032–4038, 2014.

[9] Satya Andalam and Nicola Elia. Optimal Realizable Networked Controllers for Networked Systems. *American Control Conference*, pages 336–341, 2011.

[10] Göran Andersson, Peter Donalek, Richard Farmer, Nikos Hatziargyriou, Innocent Kamwa, Prabhashankar Kundur, Nelson Martins, John Paserba, Pouyan Pourbeik, Juan Sanchez-Gasca, Ralf Schulz, Aleksander Stankovic, Charles Taylor, and Vijay Vittal. Causes of the 2003 Major Grid Blackouts in North America and Europe, and Recommended Means to Improve System Dynamic Performance. *Transactions on Power Systems*, 20 (4):1922–1928, 2005.

[11] Isaac Asimov. *Foundation*. Spectra, 1951.

[12] Mukesh Bansal, Vincenzo Belcastro, Alberto Ambesi-Impiombato, and Diego Di Bernardo. How to Infer Gene Networks from Expression Profiles. *Molecular Systems Biology*, 3(1):78, 2007.

[13] Richard Baraniuk. Compressive Sensing. *Signal Processing Magazine*, 24(4):118–124, 2007.

[14] Kenneth Burnham and David Anderson. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer-Verlag, 1998.

[15] Vasu Chetty. Necessary and Sufficient Informativity Conditions for Robust Network Reconstruction Using Dynamical Structure Functions. M.S. Thesis, Brigham Young University, 2012.

[16] Vasu Chetty, David Hayden, Sean Warnick, and Jorge Gonçalves. Robust Signal Structure Reconstruction. *Conference on Decision and Control*, pages 3184–3189, 2013.

[17] Vasu Chetty, Nathan Woodbury, Elham Vaziripour, and Sean Warnick. Vulnerability Analysis for Distributed and Coordinated Destabilization Attacks. *Conference on Decision and Control*, pages 511–516, 2014.

[18] Arne Dankers, Paul Van den Hof, Peter Heuberger, and Xavier Bombois. Dynamic Network Structure Identification with Prediction Error Methods - Basic Examples. *IFAC Symposium on System Identification*, 45(16):876–881, 2012.

[19] Chris Ding and Hanchuan Peng. Minimum Redundancy Feature Selection from Microarray Gene Expression Data. *Journal of Bioinformatics and Computational Biology*, 3(2): 185–205, 2005.

[20] Ronald Fisher. *The Design of Experiments*. Oliver & Boyd, 1960.

[21] Karl Friston, Baojuan Li, Jean Daunizeau, and Klaas Stephan. Network Discovery with DCM. *Neuroimage*, 56(3):1202–1221, 2011.

[22] Timothy Gardner Gardner, Diego Di Bernardo, David Lorenz, and James Collins. Inferring Genetic Networks and Identifying Compound Mode of Action via Expression Profiling. *Science*, 301(5629):102–105, 2003.

[23] J. Gonçalves and S. Warnick. Systems Theoretic Approaches to Network Reconstruction. In Pablo Iglesias and Brian Ingalls, editors, *Control Theory and Systems Biology*. MIT Press, 2009.

[24] Jorge Gonçalves and Sean Warnick. Necessary and Sufficient Conditions for Dynamical Structure Reconstruction of LTI Networks. *Transactions on Automatic Control*, 53(7): 1670–1674, 2008.

[25] David Grimsman, Vasu Chetty, Nathan Woodbury, Elham Vaziripour, Sandip Roy, Daniel Zappala, and Sean Warnick. A case study of a systematic attack design method for critical infrastructure cyber-physical systems. *American Control Conference*, pages 296–301, 2016.

[26] Julianne Grose, Tammy Smith, Hana Sabic, and Jared Rutter. Yeast PAS Kinase Coordinates Glucose Partitioning in Response to Metabolic and Cell Integrity Signaling. *European Molecular Biology Organization*, 26(23):4824–4830, 2007.

[27] Andras Gyorgy and Domitilla Del Vecchio. Modular Composition of Gene Transcription Networks. *PLoS Computational Biology*, 10(3):e1003486, 2014.

[28] David Hayden, Ye Yuan, and Jorge Gonçalves. Robust Network Reconstruction in Polynomial Time. *Conference on Decision and Control*, pages 4616–4621, 2012.

[29] David Hayden, Ye Yuan, and Jorge Gonçalves. Network Reconstruction from Intrinsic Noise: Non-Minimum-Phase Systems. *IFAC Proceedings Volumes*, 47(3):2329–2334, 2014.

[30] João Hespanha. *Linear Systems Theory*. Princeton University Press, 2009.

[31] Rolf Johansson. *System Modeling and Identification*. Prentice Hall, Inc, 1993.

[32] Chintan Kikani, Stephen Antonysamy, Jeffrey Bonanno, Rich Romero, Feiyu Zhang, Marijane Russell, Tarun Gheyi, Miyo Iizuka, Spencer Emtage, Michael Sauder, Benjamin Turk, Stephen Burley, and Jared Rutter. Structural Bases of PAS Domain-Regulated Kinase (PASK) Activation in the Absence of Activation Loop Phosphorylation. *Journal of Biological Chemistry*, 285(52):41034–41043, 2010.

[33] Steffi Knorn, Zhiyong Chen, and Richard Middleton. Overview: Collective Control of Multi-Agent Systems. *Transactions on Control of Network Systems*, 3(4):334–347, 2016.

[34] Andrew Lamperski and John Doyle. The Control Problem for Quadratically Invariant Systems with Delays. *Transactions on Automatic Control*, 60(7):1945–1950, 2013.

[35] Ralph Langner. Stuxnet: Dissecting a Cyberwarfare Weapon. *Security & Privacy*, 9(3): 49–51, 2011.

[36] Donatello Materassi and Murti Salapaka. On the Problem of Reconstructing an Unknown Topology via Locality Properties of the Wiener Filter. *Transactions on Automatic Control*, 57(7):1765–1777, 2012.

[37] Reza Olfati-Saber, J. Alexander Fax, and Richard Murray. Consensus and Cooperation in Networked Multi-agent Systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[38] Phil Pare. Necessary and Sufficient Conditions for State-Space Network Realization. M.s. thesis, Brigham Young University, 2014.

[39] Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2000.

[40] Karl Pearson, Alice Lee, and Leslie Bramley-Moore. Mathematical Contributions to the Theory of Evolution. VI. Genetic (Reproductive) Selection: Inheritance of Fertility in Man, and of Fecundity in Thoroughbred Racehorses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 192(1899):257–330, 1899.

[41] Jan Polderman and Jan Willems. *Introduction to Mathematical Systems Theory: A Behavioral Approach*. Springer Science & Business Media, 1998.

[42] Christopher Quinn, Negar Kiyavash, and Todd Coleman. Directed Information Graphs. *Transactions on Information Theory*, 61(12):6887–6909, 2015.

[43] Mohammad Rahimian and Victor Preciado. Detection and Isolation of Failures in Directed Networks of LTI Systems. *Transactions on Control of Network Systems*, 2(2): 183–192, 2015.

[44] Anurag Rai and Sean Warnick. A Technique for Designing Stabilizing Distributed Controllers with Arbitrary Signal Structure Constraints. *European Control Conference*, pages 3282–3287, 2013.

[45] Anurag Rai, David Ward, Sandip Roy, and Sean Warnick. Vulnerable Links and Secure Architectures in the Stabilization of Networks of Controlled Dynamical Systems. *American Control Conference*, pages 1248–1253, 2012.

[46] Wei Ren and Randal Beard. Consensus Seeking in Multi-Agent Systems Under Dynamically Changing Interaction Topologies. *Transactions on Automatic Control*, 50(5): 655–661, 2005.

[47] Michael Rotkowitz and Sanjay Lall. On Computation of Optimal Controllers Subject to Quadratically Invariant Sparsity Constraints. *American Control Conference*, pages 5659–5664, 2004.

[48] Michael Rotkowitz and Sanjay Lall. A Characterization of Convex Problems in Decentralized Control. *Transactions on Automatic Control*, 51(2):274–286, 2006.

[49] Jared Rutter, Brandon Probst, and Steven Mcknight. Coordinate Regulation of Sugar Flux and Translation by PAS Kinase. *Cell*, 111(1):17–28, 2002.

[50] Serban Sabau and Nuno Martins. Necessary and Sufficient Conditions for Stabilizability Subject to Quadratic Invariance. *Conference on Decision and Control and European Control Conference*, pages 2459–2466, 2011.

[51] Serban Sabau, Cristian Oara, Sean Warnick, and Ali Jadbabaie. Structured Coprime Factorizations Description of Linear and Time-Invariant Networks. *Conference on Decision and Control*, pages 2720–2725, 2013.

[52] Jeffrey Sachs. Blackouts and Cascading Failures of the Global Markets. *Scientific American*, 300(1):34, 2009.

[53] Michael Samoilov, Adam Arkin, and John Ross. On the Deduction of Chemical Reaction Pathways from Measurements of Time Series of Concentrations. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 11(1):108–114, 2001.

[54] Nils Sandell, Pravin Varaiya, Michael Athans, and Michael Safonov. Survey of Decentralized Control Methods for Large Scale Systems. *Transactions on Automatic Control*, 23(2):108–128, 1978.

[55] Francesca Semplici, Martine Vaxillaire, Sarah Fogarty, Meriem Semache, Amelie Bonnefond, Ghislaine Fontes, Julien Philippe, Gargi Meur, Frederique Diraison, Richard Sessions, Jared Rutter, Vincent Poitout, Philippe Froguel, and Guy Rutter. Human Mutation within Per-Arnt-Sim (PAS) Domain-Containing Protein Kinase (PASK) Causes Basal Insulin Hypersection. *Journal of Biological Chemistry*, 286(51):44005–44014, 2011.

[56] Tammy Smith and Jared Rutter. Regulation of Glucose Partitioning by PAS Kinase and Ugp1 Phosphorylation. *Molecular Cell*, 26(4):491–499, 2007.

[57] Torsten Soderstrom and Petre Stoica. *System Identification*. Cambridge University Press, 1989.

[58] Eduardo Sontag. Network Reconstruction Based on Steady-State Data. *Essays in Biochemistry*, 45(2008):161–176, 2008.

[59] Aivar Sootla and Anders Rantzer. Convenient Representations of Structured Systems for Model Order Reduction. *American Control Conference*, pages 3427–3432, 2012.

[60] Taylor Southwick. Validation of Dynamical Structure Functions for the Reconstruction of Biochemical Networks. Undergraduate honors thesis, Brigham Young University, 2011.

[61] Christian Spieth, Nadine Hassis, and Felix Streichert. Comparing Mathematical Models on the Problem of Network Inference. *Conference on Genetic and Evolutionary Computation*, pages 279–286, 2006.

[62] Siddharth Sridhar, Adam Hahn, and Manimaran Govindarasu. Cyber-Physical System Security for the Electric Power Grid. *Proceedings of the IEEE*, 100(1):210–224, 2012.

[63] André Teixeira, Saurabh Amin, Henrik Sandberg, Karl Johansson, and Shankar Sastry. Cyber Security Analysis of State Estimators in Electric Power Systems. *Conference on Decision and Control*, pages 5991–5998, 2010.

[64] André Teixeira, Daniel Perez, Henrik Sandberg, and Karl Johansson. Attack Models and Scenarios for Networked Control Systems. *Conference on High Confidence Networked Systems*, pages 55–64, 2012.

[65] Paul Van den Hof, Arne Dankers, Peter Heuberger, and Xavier Bombois. Identification of Dynamic Models in Complex Networks with Prediction Error Methods: Basic Methods for Consistent Module Estimates. *Automatica*, 49(10):2994–3006, 2013.

[66] Eugene van Someren, Bart Vaes, Wilma Steegenga, Anneke Sijbers, Koen Dechering, and Marcel Reinders. Least Absolute Regression Network Analysis of the Murine Osteoblast Differentiation Network. *Bioinformatics*, 22(4):477–484, 2006.

[67] Yuh-Shyang Wang, Nikolai Matni, and John Doyle. Localized LQR Optimal Control. *Conference on Decision and Control*, pages 1661–1668, 2014.

[68] Bartek Wilczyński and Jerzy Tiuryn. Regulatory Network Reconstruction using Stochastic Logical Networks. *Computational Methods in Systems Biology*, 4210(2006):142–154, 2006.

[69] Allen Wood and John Stankovic. Denial of Service in Sensor Networks. *Computer*, 35 (10):54–62, 2002.

[70] Enoch Yeung, Jorge Gonçalves, Henrik Sandberg, and Sean Warnick. Representing Structure in Linear Interconnected Dynamical Systems. *Conference on Decision and Control*, pages 6010–6015, 2010.

[71] Enoch Yeung, Jorge Gonçalves, Henrik Sandberg, and Sean Warnick. Mathematical Relationships Between Representations of Structure in Linear Interconnected Dynamical Systems. *American Control Conference*, pages 4348–4353, 2011.

[72] Ye Yuan, Guy-Bart Stan, Sean Warnick, and Jorge Gonçalves. Minimal Dynamical Structure Realisations with Application to Network Reconstruction from Data. *Conference on Decision and Control*, pages 4808–4813, 2009.

[73] Ye Yuan, Guy-Bart Stan, Sean Warnick, and Jorge Gonçalves. Robust Dynamical Network Structure Reconstruction. *Automatica, special issue on Systems Biology*, 47(6): 1230–1235, 2011.

[74] Zuogong Yue and Jorge Gonçalves. Dynamical Structure Functions and Granger Causality: Similarities and Differences. *Conference on Decision and Control*, pages 889–894, 2015.

[75] Kemin Zhou, John Doyle, and Keith Glover. *Robust and Optimal Control*. Prentice Hall, 1995.