2012-06-21

Analysis and Design Tools for Structured Feedback Systems

Anurag Rai
Brigham Young University - Provo

Analysis and Design Tools for Structured Feedback Systems

Anurag Rai

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

Master of Science

Sean Warnick, Chair
Daniel Zappala
Kent Seamons

Department of Computer Science

Brigham Young University

August 2012

ABSTRACT

Analysis and Design Tools for Structured Feedback Systems

Anurag Rai
Department of Computer Science, BYU
Master of Science

As we begin to analyze and construct extremely complex systems, a theory for understanding and designing the underlying architecture becomes very important. To move in the direction of a precise theory of architecture, this thesis will provide some concrete tools to analyze and design complex systems with a given network structure.

The first main result of this thesis analyzes the vulnerability of a system and shows that a system's vulnerability depends on its network structure. We will consider destabilization attacks acting on a single link in a system's logical interconnection structure. The concept of a vulnerable link is characterized, and necessary and sufficient conditions for identifying these links are provided. The vulnerability of various system architectures are then characterized by the vulnerability of their weakest link, and it is shown that every transfer function has a completely secure architecture with no vulnerable links.

The second part of this thesis focuses on synthesizing controllers with a specified network structure. It presents a new approach to distributed controller design that exploits the dynamical structure function representation of linear time invariant systems to characterize the structure of a system. The design technique sequentially constructs each link in an arbitrary controller signal structure, and the main theorem proves that either the resulting controller is stabilizing or that no controller with the desired structure can stabilize the system.

Keywords: Structured feedback systems, vulnerability, decentralized control, distributed control, signal structure.

ACKNOWLEDGMENTS

Thanks go to my adviser, Dr. Sean Warnick, and all the members of IDeA Labs.

Table of Contents

List of Figures

Chapter 1

Introduction

Scientists and engineers have started building, analyzing, and controlling large and very complex systems. For example, the Internet has become so large that we don't even have an accurate map of what it looks like. Its architecture has been criticized by several engineers and scientists and yet it provides a communication infrastructure that is a major part of the global economy. [10, 21]. Likewise, researchers in various areas of biology have been able to collect large quantities of data, but because of the extremely complex nature of the bio-molecular dynamics in the systems they study, they are only able to make limited progress in identifying the system [8]. Similarly, political scientists have collected large quantities of data on different political events happening around the globe. They have been asking whether it is possible to predict international conflict, and whether there are ways to design policies that can achieve specific goals using available data [13]. In the future, as the state of science and technology develops, it is certain that the systems that we will be dealing with will get even larger and more complex.

These systems, in general, can be thought of as entities that take inputs and produce outputs. We design systems such that they produce desired outputs for a given input. For simple systems, it rarely matters how the desired outputs are being generated. However, when the system is complex, even when the input-output results are correct, a bad implementation can make the system very difficult to maintain, or add functionalities. Moreover, it also can introduce vulnerabilities into the system, e.g. if the system is designed such that all its components depend on a single distinct component, failure of this component might cause

the whole system to collapse. Hence, having a good organization of the components of the system – the architecture – becomes crucial.

1.1 The Meaning of Architecture

Depending on the field of study, architecture may mean different things. Nevertheless, in general, the architecture of a system describes the relationship between its components. Also, a specific architecture is obtained by constraining the structure of the system.

Historically, architecture is related to the design of buildings. A building is composed of various components such as walls, windows, doors, roof, pillars, etc., and its basic functionality is to provide shelter. This functionality can be achieved by organizing the building components in various ways, and each organization gives a specific structure to the building. So, when an architect decides to constrain the structure of the building, e.g. by fixing the location of the pillars or windows or by restraining the roof to have a certain shape, it gives rise to a specific architecture.

When a software system is designed, it is divided into various components. Examples of the components might include a mechanism to store and retrieve data, a subsystem to communicate across a network, a subsystem that process the data, the user interface, etc. These components are then organized by placing constraints on their interaction. These constraints may be chosen such that the resulting architecture reduces the complexity and the software possesses different qualities such as fault-tolerance, evolvability, maintainability, etc. For example, the Model-View-Controller (MVC) architecture divides a software system into three components: i) the Model which manages the data, ii) the View which provides the user interface, and iii) the Controller which acts as a translator between the Model and the View so that the inputs from the View are fed properly to the Model and the outputs from the Model are returned properly to the View [15]. This architecture restricts direct communication between the Model and the View, allowing detached implementation of these components.

While studying networked systems, researchers sometimes use architecture synonymously with topology of the network. In [5], the authors show that some networked systems have a random architecture and others have a scale-free architecture. A scale-free network has a power-law degree distribution. It models networks in which most nodes have only a few links, held together by a few highly connected nodes. They show that the national highway network of the United States can be modeled as a random network with cities as the nodes and the highways as the links. This architecture is the result of the constraint that most cities are served by the same number of highways. Another example provided in [5] shows that the air traffic network is well modeled by a scale free network because most airports are small and are constrained to be connected to a few big airports.

In [2], the authors attempt to generalize the idea of architecture across all areas. They argue that the successful complex systems have very similar architectures, so a universal law of architecture must exist. Nevertheless, the authors are not precise about the definition of architecture in terms of concrete mathematics, and they resort to definitions like "constraints that deconstrain" to define architecture. In particular, architecture in their mind not only refers to the system structure but also the properties of the system that give rise to such structure. Using case studies in biology and engineering, they show that a good architecture has features like an "hourglass" shape or a "bow-tie" shape, which means that the architectures are usually layers of subsystems, and that there is one or only a few subsystems in the middle of the layering that provides support to a large number of subsystems on the top and the bottom. An example of such a subsystem is Legos with the property to snap. This property allows us to create a huge variety of pieces, and with these pieces, objects with various structures can be created. Moreover, regardless of the type of Legos, if it allows other pieces to snap, it can be used with other Lego objects. Another example is the hourglass shape of the Internet architecture shown in Figure 1.1 [1]. Although, a complete theory of architecture is far from sight right now, the idea that architecture is constraints on a representation of the system is a powerful abstraction of other notions of architecture.

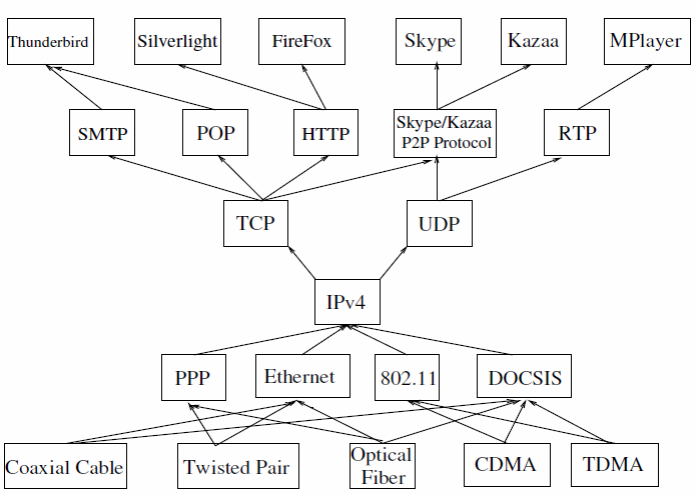


Figure 1.1: The Internet architecture has evolved into an hourglass shape [1].

1.2 Architecture of Causal, Linear Time Invariant (LTI) Systems

To study the role of architecture in determining the properties of a system precisely, in this thesis we will define architecture as constraints on a mathematical model of the system. This definition is consistent with the general idea of architecture because by constraining the model, we constrain the structure of the system. In this thesis we will consider only the class of systems called causal, Linear Time Invariant (LTI) systems. The main reason for this is because the mathematics for this class of systems is well developed and comparatively easy, and these systems are extremely useful for practical problems [11]. There are various representations to model such systems. The two most commonly used representations are the state-space model and the transfer function.

For LTI systems the state space model can be represented by the following system of linear differential equations:

$$\frac{dx}{dt} = Ax + Bu \tag{1.1}$$

$$y = Cx + Du. \tag{1.2}$$

Here u represents the inputs, x represents the states, and y represents the outputs of the system. Matrix A represents how the current state affects the next, B represents how the input affects the states, and C represents how the states are manifested in the output. The input output relation can be obtained by taking the Laplace transform of these equations, which gives

$$Y = [C(sI - A)^{-1}B + D]U.$$

This equation gives us the output, Y, of the system for any input U. $G = [C(sI - A)^{-1}B + D]$ is called the transfer function matrix. Note that there are many state space models for a particular transfer function, meaning a system can be built in different ways and still have the same input-output functionality.

The binary structure of the matrices (A, B, C) in the state-space model gives the structure and the values of the nonzero entries determine the dynamics, hence such a model describes the exact structure and dynamics in the system. On the other hand, the transfer function gives only the input-output description of the system, and does not provide any information regarding the states. It only captures the dynamics between the input and the outputs, and does not tell us how many or which states are involved to produce the output for a given input. Hence, this representation gives very limited information about the structure of the system. In fact, the only structural information it provides is whether or not an input affects a given output.

Another common representation of a system is the subsystem structure. This representation views a system as an interconnection of subsystems. Each subsystem is represented by an input-output representation, and it is assumed that each subsystem has a distinct set of states. This representation is common in engineering because engineered systems are usually designed to be modular, e.g. object oriented programming has classes with their own set of variables, the Internet protocol suite consists of layers of distinct protocols, etc.

In [12], the authors have developed a new representation for LTI systems, called the system's signal structure. This structure describes the causal dependencies between the

manifest variables of the systems and is represented using the dynamical structure function (DSF) (Q, P). Q describes the causal dependencies among the measurable states, and P represents the effect of inputs on the states. This representation is especially useful to model situations where it is not possible to measure all states, e.g. in biochemical networks. A derivation of this representation is provided in Chapter 2.

For this thesis, we will use this representation to describe the structure of the system. A binary constraint will be placed on the Q and the P matrices to describe the system's architecture in terms of its signal structure. Such a constraint restricts the structure of the system by allowing or disallowing the manifest variables of the system to affect each other. This definition of architecture is only a small and special subset of the general notion of architecture the authors are trying to describe in [2], but we hope that it will provide important insights about properties of systems, which can be used to develop a more general theory.

This thesis will solve two problems. In the first part, Chapter 3, we will analyze the *vulnerability* of a system. Vulnerability analysis considers destabilization attacks acting on a single link in its DSF representation. That is, we will study the robustness of the system to perturbations on a single link in the system. We will characterize the concept of a vulnerable link and provide necessary and sufficient conditions for identifying them. Then, we will show that all systems can have a secure or a vulnerable implementation. In the second part of this thesis, Chapter 4, we will develop a method of synthesizing controllers that have any architecture defined in the DSF representation. The design technique sequentially constructs each link in an arbitrary controller signal structure, and the main theorem proves that either the resulting controller is stabilizing or that no controller with the desired structure can stabilize the system.

Chapter 2

Dynamical Structure Function

Following [29, 30], first we will derive the DSF and present a mathematical relationship between various representations of LTI systems. We will show that the DSF provides a nice transition between the state space and the transfer function of the system, hence we will use this representation for most of our analysis.

2.1 Derivation

Let us consider a state-space LTI system

$$\begin{bmatrix} \dot{z_1} \\ \dot{z_2} \end{bmatrix} = \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} \bar{B}_1 \\ \bar{B}_2 \end{bmatrix} u \tag{2.1}$$
$$y = \begin{bmatrix} \bar{C}_1 & \bar{C}_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix},$$

where $\begin{bmatrix} \bar{C}_1 & \bar{C}_2 \end{bmatrix}$ has full row rank. This system can be transformed to:

$$\begin{bmatrix} \dot{y} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \tag{2.2}$$
$$y = \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix},$$

Here y are the states that are measured, and x are the hidden states. Now, taking Laplace Transforms of the signals in (2.2), we get

$$\begin{bmatrix} sY \\ sX \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} Y \\ X \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} U. \tag{2.3}$$

Solving for X in the second equation of 2.3 gives

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

Substituting into the first equation of (2.3) we get,

$$sY = WY + VU,$$

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 diagonal matrix with the diagonal entries of W. Then,

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

Now we can rewrite this equation as,

$$Y = QY + PU, \tag{2.4}$$

where

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

and

$$P = (sI - D)^{-1} V.$$

The matrix Q is a matrix of transfer functions where each entry Q_{ij} is a transfer function from measured state Y_j to another measured state Y_i, $i \neq j$. Also, Q is zero on the diagonal, and either zero or a strictly proper transfer function on the off diagonal. The matrix P is a matrix of zeros or strictly proper transfer functions from each input to each output without depending on any additional measured states. Together, the pair $(Q(s), P(s))$ is called the *dynamical structure function* of system (2.1).

The transfer function matrix for this system is given by

$$G = (I - Q)^{-1}P = C(sI - A)^{-1}B.$$

G_{ij} is the closed loop transfer function from input j to state i. In this thesis, we will also refer to the closed loop transfer function between manifest states. A transfer function from state j to state i is represented by H_{ij}, where

$$H = (I - Q)^{-1}.$$

Note that the transfer function from a state to an input is always zero. DSFs can also be seen as an interconnection of the systems Q and P as shown in Figure 2.1.

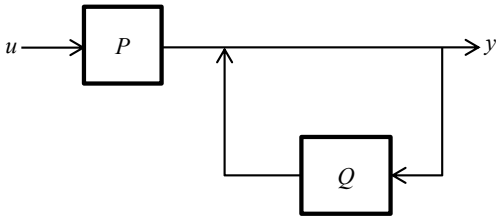


Figure 2.1: DSFs can be viewed as an interconnection of two systems characterized by the transfer function matrices Q and P, where Q is a hollow transfer function matrix. The transfer function from u to y is given by $G = (I - Q)^{-1}P$.

Definition 1. *Given a system (2.1) with the signal structure characterized by the dynamical structure function (Q,P), a* ***link*** (i, j) *of the system corresponds to any nonzero entry in P or Q.*

Example 1. *Let us consider a system with two measured states given by the following state space equation:*

$$\begin{bmatrix} \dot{y_1} \\ \dot{y_2} \\ \dot{x_1} \end{bmatrix} = \begin{bmatrix} -4 & 0 & 1 \\ 0 & -3 & 2 \\ 3 & 2 & -3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ x_1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} u \tag{2.5}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ x_1 \end{bmatrix},$$

The DSF for this system is given by,

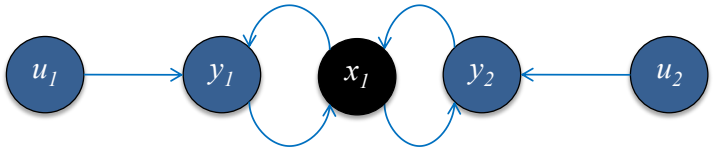
$$Q = \begin{bmatrix} 0 & \frac{2}{s^2+7\,s+9} \\ \frac{6}{(s+1)\,(s+5)} & 0 \end{bmatrix} \text{ and}$$

$$P = \begin{bmatrix} \frac{s+3}{s^2+7\,s+9} & 0 \\ 0 & \frac{1}{2\,(s+1)} + \frac{1}{2\,(s+5)} \end{bmatrix},$$

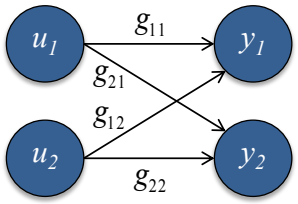
and the transfer function is given by

$$G = \begin{bmatrix} \frac{(s+1)\,(s+5)}{s^3+10\,s^2+26\,s+11} & \frac{2}{s^3+10\,s^2+26\,s+11} \\ \frac{6}{s^3+10\,s^2+26\,s+11} & \frac{s^2+7\,s+9}{s^3+10\,s^2+26\,s+11} \end{bmatrix}.$$

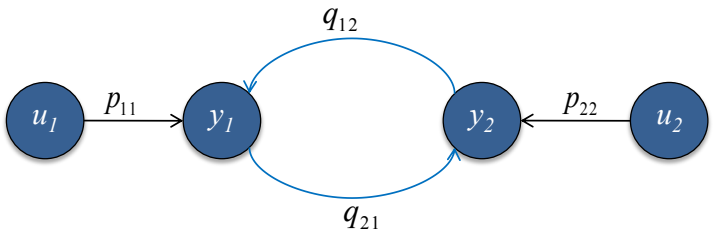
Figure 2.2 shows a graphical view of this system in various representations. Figure 2.2(a) shows the state space realization of the system, which contains information about the dependency among input, state, and output variables. Essentially, the state space of the system defines both the structure and dynamics of the entire network. A simpler transfer function representation is shown in Figure 2.2(b). This contains the dynamics of the system, but yields no information about the structure of the network. In Figure 2.2(c) the DSF of this



(a) Complete structure of the system (2.5). This can be obtained if the state space model of the system is known.



(b) Structure given by the Transfer Function of system (2.5).



(c) Structure given by the DSF of system (2.5).

Figure 2.2: Structures given by: (a) the state space, (b) the transfer function, and (c) the dynamical structure function. Here, x_1 is a hidden state: it is not measured directly and as a result is not seen in the DSF representation.

system shows the relationship between the measured states, y_1 and y_2, something not visible from the system's transfer function. In the situations when a complete state space model of the system cannot be obtained, a DSF model of the system can be used to obtain a partial structure of the system.

Notice that as the number of the measured states increases, the dynamical structure function becomes a more accurate representation of the actual state space realization. When there are no hidden states, there is a unique state space representation for the the given DSF. Also, when the hidden state is not shared between multiple links, this representation is

equivalent to the subsystem structure. Finally, when only one state is measured, Q becomes a zero matrix and P is equal to the transfer function from the inputs to the measured state.

2.2 Stability, Observability, and Controllability

There are two common notions for describing the stability of a system: the input-output stability and the internal-stability. A system is called input-output stable if it produces stable output for any stable input. A system can be tested for such stability by computing the transfer functions from its inputs to its outputs, and then testing the transfer functions' stability. But, even when a system appears to be stable using this test, there is a possibility that an internal state might be unstable and it is simply not observed in the outputs. So, we use the idea of internal stability to overcome this; a system is internally stable if all of its states are stable.

Now, we will show that the notion of internal-stability is ambiguous for DSFs. For any representation, its internal stability can be defined in terms of the smallest order state-space model that produces it. The smallest order state-space model for a given DSF is called its *structurally minimal realization*.

Definition 2. *Given a DSF (Q,P), a state space system (A,B,[I 0]) is called a structurally minimal realization if it is a system with the smallest order that generates (Q,P).*

So, a DSF is stable if its structurally minimal realization is stable. However, checking the stability of DSF using this method is not feasible. The only algorithm that we know to get a minimal realization, given in [31], is NP-complete. Also, using this method on a small system, we can see that the notion of stability is ambiguous for DSFs. In Example 2, we show that a given DSF can represent an input-output stable system that is stable or unstable internally.

Example 2. *Consider a DSF* (Q, P) *where*

$$Q = \begin{bmatrix} 0 & \frac{-1}{(s+2)} & \frac{-1}{(s+3)} \\ \frac{-1}{(s+1} & 0 & \frac{-1}{(s+3)} \\ \frac{-1}{(s+1)} & \frac{-1}{(s+2)} & 0 \end{bmatrix} \text{ and } P = \begin{bmatrix} \frac{1}{(s+4)} \\ \frac{1}{(s+4)} \\ \frac{1}{(s+4)} \end{bmatrix}.$$

Using the algorithm in [31], we find that the minimal realization has 6 states. A stable system (As,B,C) or an unstable system (Au,B,C), both with 6 states, can produce (Q,P). Here,

$$As = \begin{bmatrix} -1 & -1.00 & -1.00 & 0.50 & 1.00 & -0.75 \\ -1 & -1.00 & -1.00 & 0.00 & 1.00 & -0.75 \\ -1 & -1.00 & -1.00 & 0.50 & 0.00 & -0.75 \\ 0 & 2.00 & 0.00 & -2.00 & 0.00 & 0.00 \\ 0 & 0.00 & 2.00 & 0.00 & -3.00 & 0.00 \\ 0 & 0.00 & 0.00 & 0.00 & 0.00 & -4.00 \end{bmatrix},$$

$$Au = \begin{bmatrix} 10 & -1.00 & -1.00 & 0.50 & 1.00 & -0.75 \\ -1 & -1.00 & -1.00 & 0.00 & 1.00 & -0.75 \\ -1 & -1.00 & -1.00 & 0.50 & 0.00 & -0.75 \\ 0 & 2.00 & 0.00 & -2.00 & 0.00 & 0.00 \\ 0 & 0.00 & 2.00 & 0.00 & -3.00 & 0.00 \\ 0 & 0.00 & 0.00 & 0.00 & 0.00 & -4.00 \end{bmatrix},$$

$$B = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 4 \end{bmatrix} \text{ and } C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

This ambiguity happens only when the system represented by the DSF is uncontrollable or unobservable. If the DSF is controllable and observable, then its stability can be checked by simply checking the stability of the corresponding transfer function. Now, we characterize observability and controllability in DSF. Like transfer functions, all DSFs are observable, but they might not be controllable. This is shown in Lemma 1.

Lemma 1. *If (A,B,C) is a structurally minimal realization of a DSF (Q,P), then*

i. $(A_{22}, \begin{bmatrix} A_{21} & B_2 \end{bmatrix})$ *is controllable*

ii. (A, C) *is observable*

iii. (A, B) *might not be controllable.*

Proof. Since $W = A_{11} + A_{12}(sI - A_{22})^{-1}A_{21}$ and $V = A_{12}(sI - A_{22})^{-1}B_2 + B_1$, (A_{22}, A_{12}) must be observable and $(A_{22}, \begin{bmatrix} A_{21} & B_2 \end{bmatrix})$ must be controllable in order for the state space realization to be structurally minimal.

To test for the observability of (A,C) using the PBH test, we compute the rank of

$$\begin{bmatrix} \lambda I - A \\ C \end{bmatrix} = \begin{bmatrix} \lambda I - A_{11} & A_{12} \\ A_{21} & \lambda I - A_{22} \\ I & 0 \end{bmatrix}.$$

We can see that the first column is full rank. The second column is also full rank because (A_{22}, A_{12}) is observable. Hence (A,C) is controllable.

See Example 2 for a system that is structurally minimal but not controllable. □

To avoid the ambiguity, we will limit ourselves to the systems with a square, full-rank P matrix. In Lemma 2 we prove that systems that satisfy this assumption are always controllable. Hence, if a DSF (Q, P) satisfies this assumption, the input-output stability becomes equivalent to the internal stability. So, the test for its stability can be done by testing the stability of its transfer function $(I - Q)^{-1}P$.

Lemma 2. *If P is square and full rank then the structurally minimal realization (A, B, C) is minimal.*

Proof. Since (A, C) is observable, to prove this lemma it is sufficient to show that (A, B) is controllable, i.e. we need to show that the matrix $\begin{bmatrix} \lambda I - A & B \end{bmatrix}$ is full rank.

We have,

$$\begin{bmatrix} \lambda I - A & B \end{bmatrix} = \begin{bmatrix} \lambda I - A_{11} & A_{12} & B_1 \\ A_{21} & \lambda I - A_{22} & B_2 \end{bmatrix}.$$

From Lemma 1(i), $(A_{22}, \begin{bmatrix} A_{21} & B_2 \end{bmatrix})$ is controllable, so the bottom row is full rank.

Now we will show that $\begin{bmatrix} A_{12} & B_1 \end{bmatrix}$ has full row rank. Let us assume that $\begin{bmatrix} A_{12} & B_1 \end{bmatrix}$ is not full row rank. We know $rank(P) = rank(V) = rank(A_{12}(sI - A_{22})^{-1}B_2 + B_1)$. Since P is full rank, the rank missing in B_1 should come from $A_{12}(sI - A_{22})^{-1}B_2$. Since the rank of individual matrices is more than their product, A_{12} should have at least enough rank to make $\begin{bmatrix} A_{12} & B_1 \end{bmatrix}$ full row rank. □

Chapter 3

Vulnerable Links and Secure Architectures

The Stuxnet virus attacked an Iranian nuclear power plant in 2010 and caused the centrifuge's rotors to malfunction [28]. It gained much news coverage as the first virus attack on industrial systems. Although no serious damage was done, this attack has highlighted the necessity of improving our understanding of the security of control systems.

Researchers have predicted that attacks on industrial control systems would increase [6, 7]. As systems become more networked, securing them has become much harder and attacking them has gotten easier. In the past, securing the physical plants was enough to secure the system, but now in the networked architecture the communication channels have to be secured too. This is almost impossible to achieve, especially when these systems are being connected to the Internet, with connection features as powerful as remote access to the control centers. Regardless of the improvement in the industry's secure communications, cryptography, etc., a simple human error like someone forgetting to change a default password on their account could give an attacker complete access to the resources necessary to carry out a complicated attack. Although the security risks are real, these systems being less networked in the future is highly unlikely because of the usability advantages that a networked setting offers.

As a result, considering the security of networked control systems has become very important. A good design should make detecting attacks easy, help understand the effects of an attack, make it difficult to execute an attack, and finally minimize the consequences if an attack is successful. We will contribute in designing more secure networked systems by

identifying architectures which make a link completely secure against attacks that attempt to destabilize the system. Our result also gives a measure of link vulnerability, which corresponds to the minimum size of a destabilizing attack on the link. This can be a useful tool in understanding the security of a networked system.

In this thesis, we view security as a robustness issue and focus on making systems robust against perturbations on a single communication channel. First, we give a summary of the types of attacks that a system might suffer. Then, in Section 3.3 we present our main result. Finally in Section 3.4 we give some examples to illustrate the applications of our theory.

3.1 Attack Models

In the literature, attacks on control systems have been classified into two types: *denial of service attacks*, when the attacker jams a channel in order to destabilize the system, and *deception attacks*, when the attack adds perturbations on particular links in order to compromise the reliability of the controller's state estimates [3]. We consider a hybrid attack model where the attacker adds perturbations to the channels in order to destabilize the system. We call this type of attack a *destabilizing attack*.

3.1.1 Denial of Service (DoS) Attack

Denial of service attacks prevent signals from reaching their intended destination. This is probably the easiest and most common attack, and it is modeled as removal of an edge in an interconnected structure. It might be done by jamming the communication channel, disrupting the transmitter/receiver, changing the routing protocol, saturating the receiver with extraneous signals, etc. The attacker's intent of such an attack could be to degrade the system's performance or to completely destabilize the system. [18] shows that performance of networked control systems could decrease significantly under a DoS attack. [3] gives a

method to find an optimal controller that minimizes the effect of such an attack on linear control systems.

In [20], the authors study whether a DoS attack on certain links can make the system unobservable or uncontrollable. In a feedback system, the plant needs to be observable and controllable in order to be stabilized by the controller. They also develop graph theoretic algorithms to identify the minimal number of edges which are necessary for preserving controllability and observability.

3.1.2 Deception Attack

The goal of a deception attack is to change the state estimates computed by a model-based controller. This type of attack is modeled as a stable additive perturbation to an edge in the network. All stabilizing controllers make the closed loop system stable, hence, a stabilizing controller is necessarily stabilizable from the plant. So, if an attacker gains access to the communication channel between the plant and the controller, state estimates of a model-based controller can be altered. To prevent this, many real systems such as power systems, sensor networks, etc., are equipped with a Bad Data Detector (BDD) [17, 19, 24]. A BDD, using the model of the plant, detects deviation of the state estimates from the expected and raises an alarm to notify the human operator. Because of the presence of measurement noise, this deviation is never zero, so the BDD ignores deviations that are smaller than a specified threshold. Hence, in the presence of BDDs, the attack has to change the state estimates without increasing the chance of raising an alarm.

In [17] the authors study this kind of attack in the context of a power system. They show that it is in fact possible for an attacker to change the state estimates to a specific value without increasing the chance of being detected. [19] studies a similar problem in the scenario of a wireless sensor network. This paper produces an approximation of the set of all the possible values the attacker could drive the estimates to.

[24] studies a slightly different problem. Here, the goal of the attacker is to change the estimate of one of the states without increasing the chance of being detected. The authors recognize that while doing this the attacker might want to use the fewest channels possible or might try to keep the magnitude of the attack signal small. For each type of attack, the authors then give a formulation of a *security index* of the system.

3.1.3 Destabilizing Attack

Like deception attacks, these attacks effectively arise as an additive perturbation on a link in the system interconnection structure. Unlike deception attacks, however, they seek to destabilize the system rather than simply move the system state to a desired value without being detected. BDDs are clearly capable of detecting the destabilization resulting from such attacks, nevertheless serious damage and even complete plant shut-down may occur as a result of the attack.

A rich literature in systems and control theory explores the destabilization of systems due to additive perturbations, see for example [11] and the references therein. Security analysis of destabilizing attacks thus appears to be a robustness problem with respect to certain classes of perturbations. Indeed, we adopt this point of view, and consider security problems to be essentially robustness problems of various types.

The contribution of this work, applied to this class of attacks, is in the solution of a certain class of robustness problems over a particular kind of link model–corresponding to logical, rather than the physical, links of a system–and with respect to a specific class of perturbations. Unlike standard system robustness measures that generally consider destabilizing perturbations acting over all channels and nodes of a system, here we restrict our attention specifically to perturbations that disrupt a single link in the system's signal structure. Our analysis then considers such single-link perturbations over all possible system links. In the next section we explore our link model in detail.

3.2 Link Models

The destabilizing attacks considered here are additive perturbations acting on a single link in a system's logical interconnection structure. There are many characterizations of a system's structure, see for example [29, 30]. One characterization would consider the interconnection structure among subsystems. This definition of structure, also called the system's subsystem structure, would represent the physical interconnection between physical components of a particular networked system. Under this notion of structure, a *link* would represent the signal passing between two subsystem nodes within the subsystem interconnection architecture. In contrast to the subsystem structure, this work considers another definition of system structure and, consequently, a different notion of a system link.

In this work, we consider a partition on signals of the system into two categories: exposed signals and hidden signals. The logical interconnection structure, or architecture–also called the system's signal structure–is the causal relationship between exposed signals in the system. In this definition of structure, a *link* is a system describing the causal dependency between two exposed signal nodes of the logical interconnection architecture. We will use DSF to represent the signal structure of a system.

Some important consequences of this definition of a link include the fact that a link may represent a very indirect and complicated pathway–through various hidden signals that may be components of other links in the system. Thus a link is associated with a particular set of dynamics–a system–that characterizes how the input signal is transformed into the output signal. The fact that hidden signals may be shard between links, however, is an important distinction between signal and subsystem interconnection structures. Note that a state of one subsystem, interconnected with others in a subsystem architecture (such as a standard feedback interconnection between two blocks), is never shared with other subsystems; the subsystem architecture effectively partitions the states of the networked system. In contrast, states on the links of the signal structure may, in fact, be shared with those of other links. This degree of abstraction is important for security problems because an additive perturbation

on a link of the signal structure does not represent the corruption of a particular channel, as it would in the subsystem structure, but rather the idea that an attacker infiltrated a particular dependency between specific manifest variables.

3.3 Vulnerable Links

In this work, vulnerability refers to the destabilization of a system resulting from the corruption of a single link in its signal architecture. We begin with a definition of a vulnerable link.

Definition 3. *Given a system 2.1 with signal structure characterized by the dynamical structure function (P,Q), a link in (P,Q) is called* vulnerable *if there exists a stable perturbation on the link that makes the system unstable.*

Example 3. *Let us consider a system with*

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

This system is stable because the transfer function,

$$G = \frac{1}{s^2+4s+3} \begin{bmatrix} s+2 & 1 \\ 1 & s+2 \end{bmatrix},$$

is stable. Now let us add a perturbation $\Delta = \frac{3}{s+2}$ to the link Q_{12} as shown in Figure 3.1. The resulting transfer function is

$$\bar{G} = \frac{1}{s(s+4)} \begin{bmatrix} s+2 & 1 \\ 4 & s+2 \end{bmatrix},$$

which is unstable. Hence the link Q_{12} is a vulnerable link. Similarly, it can be shown that Q_{21} is vulnerable, although neither P_{11} nor P_{22} are vulnerable.

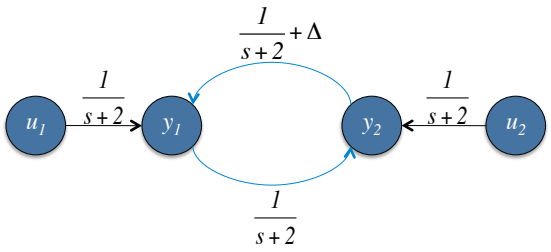


Figure 3.1: The system with the perturbation Δ. Black arrows indicate secure links, while blue arrows indicate vulnerable links.

3.3.1 Condition for Vulnerability

Given that an attacker has the knowledge of the dynamical structure function representation of a system, we will derive a necessary and sufficient condition for a link to be vulnerable.

Theorem 1. *Let us consider a stable system (P, Q). There exists a stable additive perturbation Δ on a link from node i to node j, either in P or Q, that makes the system unstable if and only if the closed loop transfer function from node j to i is nonzero.*

Proof. The system with the perturbation Δ can be represented as the linear fractional transformation in Figure 3.2, where T is the associated closed loop transfer function, and w_i, w_j represent the signals at node i and j respectively. This system is stable if and only if the system in Figure 3.3 is stable (see [11]). If $T_{ij} = 0$, any stable Δ does not affect the stability of the system in Figure 3.3. Thus the closed loop system in Figure 3.2 is stable for all Δ.

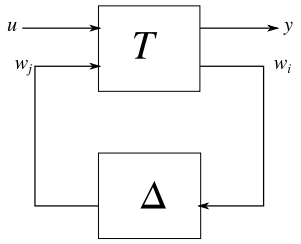


Figure 3.2: System with the perturbation $\Delta e_i e_j^T$

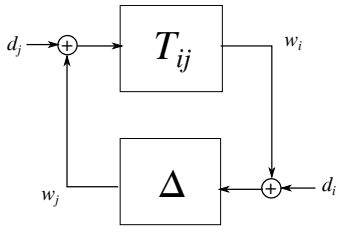


Figure 3.3: Necessary and sufficient condition for stability of the system in Figure 3.2

If $T_{ij} \neq 0$, then the system in Figure 3.3 is unstable if any of the transfer functions of $\begin{bmatrix} d_j \\ d_i \end{bmatrix} \rightarrow \begin{bmatrix} w_j \\ w_i \end{bmatrix}$ is unstable. We have,

$$w_j = \frac{1}{1 - T_{ij}\Delta} \begin{bmatrix} T_{ij}\Delta & \Delta \end{bmatrix} \begin{bmatrix} d_j \\ d_i \end{bmatrix}.$$

Let $T_{ij} = \frac{N}{D}$ and $\Delta = \frac{\delta_N}{\delta_D}$, then

$$w_j = \frac{D\delta_D}{D\delta_D - N\delta_N} \begin{bmatrix} \frac{N\delta_N}{D\delta_D} & \frac{\delta_N}{\delta_D} \end{bmatrix} \begin{bmatrix} d_j \\ d_i \end{bmatrix}. \tag{3.1}$$

For a polynomial to be stable it is necessary that all its coefficients are of the same sign. In the case of the polynomial

$$R(s) = D\delta_D - N\delta_N, \tag{3.2}$$

it is easy to see that a properly designed Δ can zero out at least one of the terms. Thus, there exists a Δ that destabilizes these transfer functions. □

Note that when we are considering the vulnerability of the links in Q, $T = H = (I - Q)^{-1}$, gives the closed loop transfer functions. Now, we will present some implications of this result.

Corollary 1. *None of the links in P are vulnerable.*

Proof. This is true because the transfer function from the states to the input is always zero. □

Corollary 2. *If T_{ij} is nonzero, there exists a perturbation $\Delta \in \mathbb{R}$ that destabilizes the system in Figure 3.3.*

Proof. Let $\Delta \in \mathbb{R}$, $l_{ij} = \frac{N_l}{D_l}$. Thus, $\frac{\delta_n}{\delta_d} = \frac{\Delta D_l + N_l}{D_l}$, and the polynomial in (3.2) becomes $D_l D - N(D_l \Delta + N_l)$. We can see that at least one of the terms in this polynomial can be zeroed out by choosing appropriate Δ, making the polynomial unstable. □

Corollary 3. *Let us consider a stable system,*

$$\begin{aligned} \dot{x} &= Ax + Iu, \\ y &= Ix, \end{aligned} \tag{3.3}$$

where $A \in \mathbb{R}^{n \times n}$ and let $G = (sI - A)^{-1}$. There exists a perturbation $K = \Delta e_i e_j^T, \Delta \in \mathbb{R}$, such that $(A + K)$ is not Hurwitz, if and only if the transfer function from input u_i to output y_j, G_{ji}, is nonzero.

Proof. If the perturbation is on the diagonal entry of A, then it is easy to see that a destabilizing perturbation always exists and G_{ii} is never zero. Let $D = diag(A_{11}, A_{22}, ..., A_{nn})$. The dynamical structure function of the system is given by $P = (sI - D)^{-1}$ and $Q = (sI - D)^{-1}(A - D)$. Any perturbation $K = \Delta e_i e_j^T, i \neq j$ affects only the link Q_{ij}. Hence, the perturbation can make the system unstable if and only if the transfer function H_{ji} is nonzero. Also, the diagonal entries of P are nonzero, and $G = HP$. Thus, the transfer function H_{ij} is nonzero if and only if G_{ji} is nonzero.

□

Example 4. *Let us consider a system of the form 3.3 with*

$$A = \begin{bmatrix} -1 & 0 & -4 & 3 \\ 2 & -2 & 0 & 0 \\ 3 & 0 & -2 & -4 \\ 0 & 3 & -2 & -5 \end{bmatrix}.$$

Here the eigenvalue of A are $\sigma = \{-1.5000 + 3.4278j, -1.5000 - 3.4278j, -6.7016, -0.2984\}$. Hence, the system is stable. In this system, the link from x_4 to x_1 is not vulnerable because $G_{41} = 0$. Notice that this example is not a trivial example, like a diagonal or a triangular system, since there are cycles that contain both nodes x_1 and x_4.

Corollary 4. *Let $A \in \mathcal{R}^{n\times n}$. A perturbation on the $(i,j)^{th}$ entry of A changes its eigenvalues if and only if the $G_{ji} \neq 0$, where $G = (sI - A)^{-1}$ is the transfer function matrix i.e. the (i,j) minor of $(sI - A)$ is nonzero.*

Proof. Take the system from Corollary 3. We can see that a perturbation on the $(i,j)^{th}$ entry has no effect on the system if $G_{ij} = 0$. Also, if $G_{ji} \neq 0$, the perturbation forms a closed loop system, such as the one given in Figure 3.3, in which case Δ definitely changes the poles of the system. □

If we take the A matrix from Example 4, note that its eigenvalues stay unchanged for any perturbation on the $(1,4)^{th}$ entry.

3.3.2 Structure and Vulnerability

To perform the vulnerability analysis of a system, we assume that the attacker can only modify existing links and cannot create new links. With this assumption, we can see that systems where the output nodes do not form a cycle are always secure, because in such a case the nodes can be permuted to obtain a triangular Q matrix. A triangular Q matrix gives a triangular H, and by applying Theorem 1 we can see that all the existing links are

secure. Note that the existence of secure links doesn't always mean they are from a triangular system. For example, the link Q_{14} is secure in the system given in Figure 3.4, which is the signal structure architecture of the state-space system in Example 4.

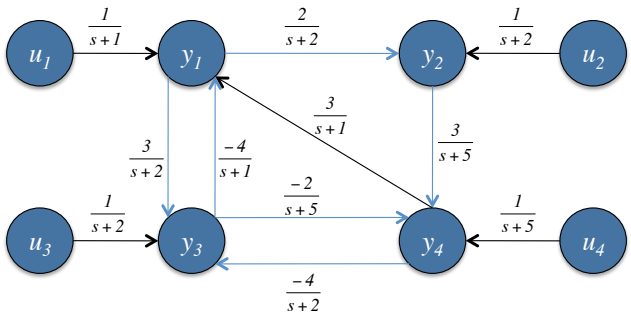


Figure 3.4: A system with a secure link in a cycle. Black arrows represent the secure links.

Noting that certain graphical structures result in secure links begs the question of whether there are particular dynamics that contribute to secure or vulnerable links in the system's architecture. The following theorem answers this question.

Theorem 2. *Every transfer function G has a completely secure architecture $(\bar{P}, \bar{Q})$.*

Proof. For any transfer function G, note that $(P = G, Q = 0)$ is an admissible Dynamical Structure Function since $G = (I - 0)^{-1}G$. From Corollary 1, we see that none of the links in P are vulnerable, and since Q has no links, the system is secure. □

This result shows that the vulnerability of a system is structure dependent and not a function of the system dynamics. This fact highlights one difference between the vulnerability, which depends on the system structure and not the dynamics, and the robustness, which depends on the dynamics and not the system structure.

3.3.3 Measure of Vulnerability

Feedback is very common in both natural and engineered systems. Nevertheless, such structures usually generate vulnerable links. Thus, a measure of vulnerability is essential to understand the security of the system.

Given a signal architecture (P, Q) with associated closed loop transfer function T, the vulnerability of link (i, j) is given by

$$v_{ji} = ||T_{ij}||_\infty, \tag{3.4}$$

which is the inverse of the smallest perturbation required on link (i, j) to destabilize the system. Since all the links in P are secure, we only consider the links in Q while computing the vulnerability, hence, $T = H$. The vulnerability of the system is given by

$$V = \max_{(i,j)\in Q} v_{ji} \tag{3.5}$$

$$= \max_{(i,j)\in Q} ||T_{ij}||_\infty \tag{3.6}$$

This measure allows us to associate a size of the smallest destabilizing perturbation with every link in the system architecture. Secure links thus have a vulnerability of 0. Note that V, the system vulnerability, is less than or equal to the inverse of the size of the smallest destabilizing perturbation for the system, since link perturbations are restricted to act on a single link only.

3.4 Numerical Example

Let us consider a system with the architecture given in Figure 3.5(a) where,

$$P = \begin{bmatrix} \frac{1}{s+1} & 0 & 0 \\ 0 & \frac{1}{s+1} & 0 \\ 0 & 0 & \frac{1}{s+1} \end{bmatrix}$$

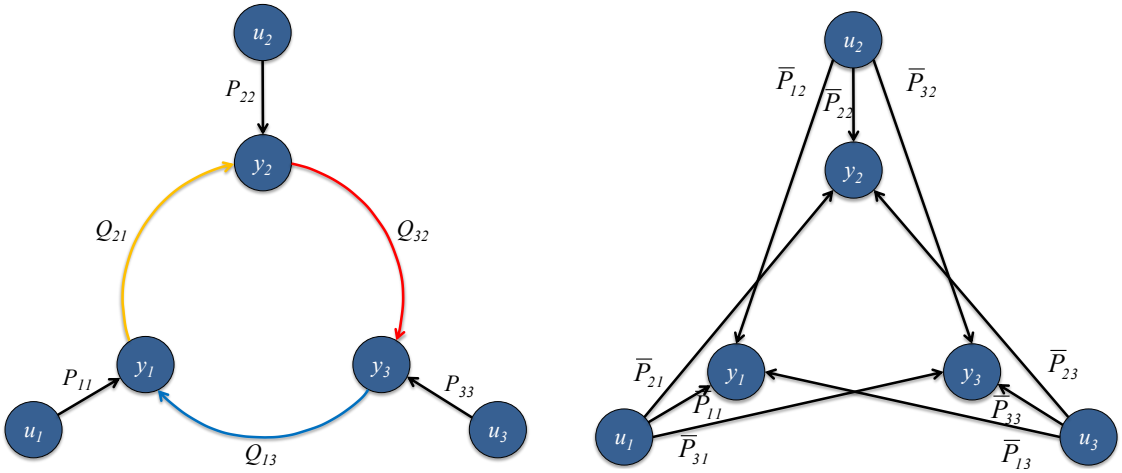
and

$$Q = \begin{bmatrix} 0 & 0 & \frac{1}{s+1} \\ \frac{1}{s+2} & 0 & 0 \\ 0 & \frac{1}{s+3} & 0 \end{bmatrix}.$$

The transfer function matrix for the system is given by

$$G = \begin{bmatrix} \frac{s^3+6s^2+11s+6}{d(s)} & \frac{s+2}{d(s)} & \frac{s^2+5s+6}{d(s)} \\ \frac{s^2+4s+3}{d(s)} & \frac{s^3+6s^2+11s+6}{d(s)} & \frac{s+3}{d(s)} \\ \frac{s+1}{d(s)} & \frac{s^2+3s+2}{d(s)} & \frac{s^3+6s^2+11s+6}{d(s)} \end{bmatrix},$$

where $d(s) = s^4 + 7s^3 + 17s^2 + 16s + 5$. By the small gain theorem, the size of the smallest destabilizing perturbation is $||G||_\infty^{-1} = 0.42$.



(a) Vulnerable architecture (b) Completely secure architecture

Figure 3.5: Vulnerable and secure architectures for the same transfer function. Black links are secure, vulnerable links are colored blue, yellow, and red in the increasing order of their vulnerability.

Let $H = (I - Q)^{-1}$ represent the transfer function between the measured states y_i. Since the links in P are not vulnerable, we consider the perturbations on the links in Q which are the links (y_1, y_2), (y_2, y_3), and (y_3, y_1). To compute the vulnerability of these links we

need the following transfer functions:

$$H_{12} = \frac{s+2}{s^3+6s^2+11s+5}$$
$$H_{23} = \frac{s+3}{s^3+6s^2+11s+5}$$
$$H_{31} = \frac{s+1}{s^3+6s^2+11s+5}.$$

For this system $v_{21} = 0.4$, $v_{32} = 0.6$, and $v_{13} = 0.2$. Hence, $V = v_{23} = 0.6 < ||G||_\infty$, and the smallest perturbation on a single link that can destabilize this system must have a gain of $\frac{1}{V} = 1.67$.

This system can also be implemented as shown in Figure 3.5(b), where $\bar{P} = G$. This is one of the secure implementations of the system in Figure 3.5(a). Note that in practice, it is not always possible to change the structure of the system to the one in Figure 3.5(b). Changing the structure like this requires a complete re-implementation of the system, which might not be allowed. In such cases, the architecture for the portion of the system that needs to be designed (the controller) can be designed in order to minimize the vulnerability using the metric presented in the previous section. In the next chapter we will develop a method to construct stabilizing controllers with a particular signal structure.

From this example we thus observe the following:

- The same transfer function can exhibit both vulnerable and secure architectures,
- System robustness, characterized by the size of the smallest destabilizing perturbation (0.42 in this example), is not equivalent to the inverse of the system vulnerability, characterized by the size of the smallest destabilizing perturbation on a single link (about 1.67 in this example),
- Only links in Q can be vulnerable.

Chapter 4

Synthesis of Structured Controllers

Distributed controller design concerns the imposition of architectural constraints on a feedback controller while attempting to stabilize, and possibly optimize, the closed-loop performance of a given system, called the *plant*. The problem only arises when the plant is multi-input and multi-output, and the standard notion of architectural constraints implies that certain elements of the controller transfer function matrix are forced to be zero.

Although the sparsity pattern of a transfer function is certainly one notion of a system's structure, it is typically the weakest form of system structure considered. There are other notions of system structure, such as the interconnection pattern of subsystems or the sparsity pattern of a state space realization that are stronger structural concepts [29, 30]. Here we say they are stronger structural concepts in the sense that the interconnection of subsystems or a particular state space realization determines the sparsity pattern of the associated transfer function, but not the other way around.

In this chapter we consider the signal structure of the system, given by the DSF, that is both stronger than the sparsity pattern of the transfer function but weaker than the sparsity pattern of the system's state space realization. If we use these two system representations as extremes, suggesting that the sparsity pattern of the state realization is the *complete computational structure* of the system while the sparsity pattern of the transfer function may contain little (if any) structural information, then the signal structure is squarely between the two in terms of its structural informativity. The system's DSF describes the *open-loop* causal dependencies among manifest variables (inputs and outputs), whereas the transfer

function describes the *closed-loop* dependencies from inputs to outputs. Thus, while a DSF may be intricately structured, its corresponding transfer function may be fully connected, essentially exhibiting no particular structure (see Figure 1). This is why many interesting distributed control problems are not described well by imposing sparsity constraints on the controller's transfer function.

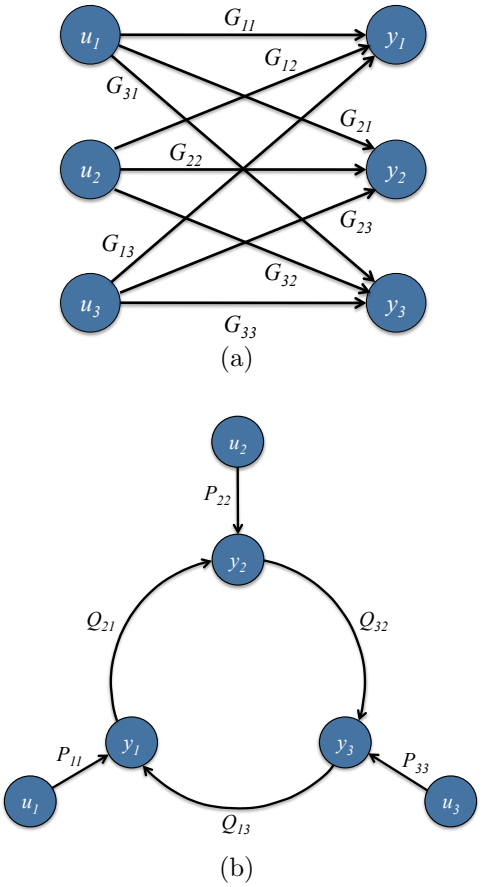


Figure 4.1: Two distinct notions of structure for the same system. The top figure indicates that the transfer function, evidently a 3×3 matrix $G(s)$, is full and unstructured, while the bottom figure indicates that the signal structure, represented by the dynamical structure function with two 3×3 matrices $Q(s)$ and $P(s)$ where $G(s) = (I - Q(s))^{-1}P(s)$, is sparse and definitively structured. Note that the bottom figure may represent communication links, and since there is a pathway from every input to every output, the associated transfer function may be full, as in the top figure.

This chapter describes a technique for designing stabilizing controllers with a particular signal structure for a given plant, or demonstrating that no such controller exists. The next section discusses related work, while the following section details mathematical preliminaries

regarding dynamical structure functions as a partial structure representation of linear time invariant systems. We then present the design procedure and the main result, which proves that the design procedure delivers a stabilizing controller with the desired structure if possible. Examples and conclusions follow.

4.1 Related Work and Background

One of the first results on the existence of a decentralized controller was given in [27]. It developed the idea of *fixed modes* and showed that a decentralized controller exists if and only if the system had no unstable fixed modes. More precisely, it showed that a system (A, B, C) is stabilizable with a diagonal or block diagonal controller K if and only if $A - BKC$ does not have any unstable eigenvalues that cannot be moved by changing the nonzero entries of K. This result was extended in [25] by showing that this is in fact true for any distributed controller K, not just for diagonal and block diagonal. The authors also present methods to synthesize the decentralized stabilizing controller.

In [22] the authors show that if the structure of the transfer function matrices of the plant and the controller meet a certain condition, known as the *quadratic invariance* condition, then the problem of synthesizing the optimal controller is convex. In [16] the authors show that the quadratic invariance condition is necessary and sufficient for the problem of synthesizing the optimal controller to be convex. This method requires a decentralized stabilizing controller to initialize the convex optimization problem, so to complete the process, an algorithm to obtain such a controller is provided in [23].

A different type of distributed controller design has been proposed in [26]. The approach taken in this paper enforces the controller to have the same network structure as the plant. The structure in this paper is defined as the constraint on the interconnection of sub-systems, or the subsystem structure. Hence, the plant and the controller can share the same communication network reducing the implementation cost. An algorithm to synthesize a sub-optimal controller with such structure is also provided in this paper.

In this work we introduce a similar, but a more general controller design problem. Instead of the controller having to have the same structure as the plant, we allow it to have any structure. Also, the structure is defined as a constraint on the signal structure. In Figure 4.2 we show an example of a plant and a corresponding controller structure that we might want to have. When a controller has such a structure, we can see that all the controller units affect each other directly or indirectly, hence, the controller transfer function matrix is completely full. As a result, using the usual approach of placing binary constraints on the controller transfer function will produce a centralized controller as shown in 4.3. Also, most of these setups do not meet the quadratic invariance criterion. These issues are illustrated in Example 5.

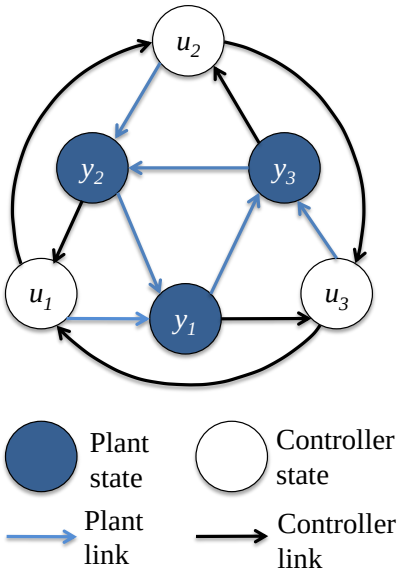


Figure 4.2: Plant with the signal structure as in Figure 4.1(b) interconnected with controller with a particular desired distributed structure.

In this chapter, the structure of a controller is defined as a sparsity constraint on the Q matrix; we assume, for the ease of exposition, that the P matrix is diagonal. We will use the binary matrices (Q^{bin}, P^{bin}) to represent the sparsity of the desired controller. The $(i,j)^{th}$ element of Q^{bin}, $q_{ij}^{bin} = 1$ if the j^{th} controller unit can communicate with the i^{th} controller unit. Similarly, $p_{ij}^{bin} = 1$ if the j^{th} plant unit communicates with the i^{th} controller unit. K^{bin} represents a structural constraint on the transfer function of the controller.

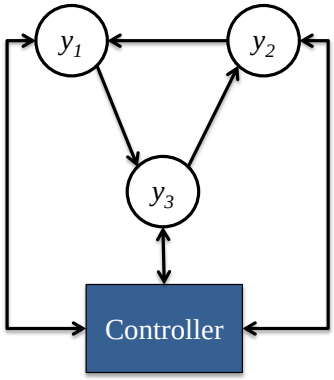


Figure 4.3: Since the desired signal structure for the controller in Figure 4.2 yields a full transfer function, other design methods yield a centralized controller.

Example 5. *Using this notation, the desired controller in Figure 4.2 is given by:*

$$P^{bin} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and

$$Q^{bin} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Let us assume that the transfer function $Q_{ij} = q_{ij}$ if $Q_{ij}^{bin} = 1$, and $Q_{ij} = 0$ otherwise, and similarly $P_{ij} = p_{ij}$ if $P_{ij}^{bin} = 1$, and $P_{ij} = 0$ otherwise. The corresponding transfer function matrix for this controller is given by, (Q^{bin}, P^{bin})

$$\begin{aligned} K &= (I - Q_k)^{-1} P_k \\ &= \begin{bmatrix} -\frac{p_{11}}{q_{13}\, q_{21}\, q_{32} - 1} & -\frac{p_{12}\, q_{13}\, q_{32}}{q_{13}\, q_{21}\, q_{32} - 1} & -\frac{p_{13}\, q_{13}}{q_{13}\, q_{21}\, q_{32} - 1} \\ -\frac{p_{11}\, q_{21}}{q_{13}\, q_{21}\, q_{32} - 1} & -\frac{p_{12}}{q_{13}\, q_{21}\, q_{32} - 1} & -\frac{p_{13}\, q_{13}\, q_{21}}{q_{13}\, q_{21}\, q_{32} - 1} \\ -\frac{p_{11}\, q_{21}\, q_{32}}{q_{13}\, q_{21}\, q_{32} - 1} & -\frac{p_{12}\, q_{32}}{q_{13}\, q_{21}\, q_{32} - 1} & -\frac{p_{13}}{q_{13}\, q_{21}\, q_{32} - 1} \end{bmatrix} \end{aligned}$$

We can see that this transfer function matrix is full, hence this controller cannot be obtained by placing binary constraints on the transfer function matrix.

Quadratic Invariance results presented in [22] provide a method to place other types of constraints on the transfer function. For the structure given in this example the constraints are as follows:

$$\frac{k_{21}}{k_{11}} = \frac{k_{32}}{k_{13}}, \frac{k_{31}}{k_{21}} = \frac{k_{32}}{k_{22}}, \text{ and } \frac{k_{12}}{k_{32}} = \frac{k_{13}}{k_{33}} \tag{4.1}$$

Let us assume that plant has the structure as shown in Figure 4.2. If $\bar{p}_{ij}$ and $\overline{q}_{ij}$ represents the transfer functions on the DSF of the plant, the transfer function matrix for the plant is given by

$$G = \begin{bmatrix} -\frac{\bar{p}_{11}\bar{q}_{12}\bar{q}_{32}}{\bar{q}_{12}\bar{q}_{31}\bar{q}_{32}-1} & -\frac{\bar{p}_{22}}{\bar{q}_{12}\bar{q}_{31}\bar{q}_{32}-1} & -\frac{\bar{p}_{33}\bar{q}_{12}}{\bar{q}_{12}\bar{q}_{31}\bar{q}_{32}-1} \\ -\frac{\bar{p}_{11}\bar{q}_{32}}{\bar{q}_{12}\bar{q}_{31}\bar{q}_{32}-1} & -\frac{\bar{p}_{22}\bar{q}_{31}\bar{q}_{32}}{\bar{q}_{12}\bar{q}_{31}\bar{q}_{32}-1} & -\frac{\bar{p}_{33}}{\bar{q}_{12}\bar{q}_{31}\bar{q}_{32}-1} \\ -\frac{\bar{p}_{11}}{\bar{q}_{12}\bar{q}_{31}\bar{q}_{32}-1} & -\frac{\bar{p}_{22}\bar{q}_{31}}{\bar{q}_{12}\bar{q}_{31}\bar{q}_{32}-1} & -\frac{\bar{p}_{33}\bar{q}_{12}\bar{q}_{31}}{\bar{q}_{12}\bar{q}_{31}\bar{q}_{32}-1} \end{bmatrix}.$$

By computing the product $Z = KGK$ we can see that

$$\frac{z_{21}}{z_{11}} \neq \frac{z_{32}}{z_{13}}.$$

This violates the constraints given in Equation (4.1), hence, the plant and the controller are not quadratically invariant and the algorithm in [22] cannot be used to construct such controllers.

In [14], [9], etc., sequential design methods have been used to construct decentralized controllers. Although these methods do not produce the optimal controller, they provide an efficient method to synthesize a nominal stabilizing controller with a desired decentralized sparsity pattern in its transfer function. We will use a similar strategy to design a stabilizing controller with constraints on the signal structure in Section 4.2. In the event that this process cannot produce a stabilizing controller, we will show that there is no controller of the given signal structure that stabilizes the plant.

4.2 Main Result

In this section, we present a procedure to design a controller (Q, P) with a structure given by (Q^{bin}, P^{bin}) to stabilize a plant with the transfer function matrix G. The procedure is as follows:

Procedure $\mathbb{P}$

1. Choose an undesigned link p_{ij} such that $p_{ij}^{bin} = 1$
2. Design p_{ij} to stabilize g_{ji} such that there is no pole zero cancellation in PG. That is, the controller link is designed such that it stabilizes the transfer function it sees, and there is no pole-zero cancellation.
3. After adding p_{ij}, if the closed loop system (G, P) is still unstable, repeat for all p_{xy}, $p_{xy}^{bin} = 1$.
4. If the closed loop system S, formed by adding P in feedback with G, is still unstable, add links in Q^{bin} such that there is no pole-zero cancellation between Q and S.

Theorem 3. *Given a transfer function matrix, G, and a desired signal structure for a feedback controller characterized by (Q^{bin}, P^{bin}), Procedure $\mathbb{P}$ either delivers a stabilizing controller with the desired structure or no such controller exists.*

This theorem says that if the controller obtained using this procedure does not stabilize the plant, then there is no controller of the given structure that can stabilize it. Hence, this procedure provides a test for the existence of a structured stabilizing controller, and if such a controller exists, it synthesizes a nominal stabilizing controller that meets the structural constraint. Before proving this theorem, we will prove some lemmata.

Lemma 3. *Let K be the controller transfer function. A link k_{ij} cannot affect a mode of the plant G that is not observable or controllable from this link.*

Proof. Let,

$$G = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \text{ and } k_{ij} = \left[\begin{array}{c|c} A_k & B_k \\ \hline C_k & 0 \end{array}\right].$$

Since we are only adding one link, both of these systems are SISO. Using the Kalman decomposition on G, we can transform it such that

$$A = \begin{bmatrix} A_{co} & 0 & A_{\times o} & 0 \\ A_{c\times} & A_{c\bar{o}} & A_{\times\times} & A_{\times\bar{o}} \\ 0 & 0 & A_{\bar{c}o} & 0 \\ 0 & 0 & A_{\bar{c}\times} & A_{\bar{c}\bar{o}} \end{bmatrix}, B = \begin{bmatrix} B_{co} \\ B_{\bar{c}o} \\ 0 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} C_{co} & 0 & C_{c\bar{o}} & 0 \end{bmatrix}, \text{ and } D = d.$$

Here, the eigenvalues of $A_{c\bar{o}}$, $A_{\bar{c}o}$, and $A_{\bar{c}\bar{o}}$ are the modes of G that are unobservable, uncontrollable, and both respectively from feedback link k_{ij}.

The closed loop modes are given by the eigenvalues of the following matrix:

$$A_{cl} = \begin{bmatrix} A & BC_k \\ B_kC & A_k + B_kDC_k \end{bmatrix}$$

$$= \begin{bmatrix} A_{co} & 0 & A_{\times o} & 0 & B_{co}C_k \\ A_{c\times} & A_{c\bar{o}} & A_{\times\times} & A_{\times\bar{o}} & B_{c\bar{o}}C_k \\ 0 & 0 & A_{\bar{c}o} & 0 & 0 \\ 0 & 0 & A_{\bar{c}\times} & A_{\bar{c}\bar{o}} & 0 \\ B_kC_{co} & 0 & B_kC_{c\bar{o}} & 0 & A_k + B_kDC_k \end{bmatrix}$$

Transforming this matrix using the permutation

$$T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix},$$

we get,

$$\begin{aligned} A_{clT} &= TA_{cl}T' \\ &= \begin{bmatrix} A_{c\bar{o}} & A_{c\times} & B_{c\bar{o}}C_k & A_{\times\bar{o}} & A_{\times\times} \\ 0 & A_{co} & B_{co}C_k & 0 & A_{\times o} \\ 0 & B_kC_{co} & A_k + B_kDC_k & 0 & B_kC_{c\bar{o}} \\ 0 & 0 & 0 & A_{\bar{c}\bar{o}} & A_{\bar{c}\times} \\ 0 & 0 & 0 & 0 & A_{\bar{c}o} \end{bmatrix} \end{aligned}$$

We can see that A_{clT} is block triangular, and the uncontrollable or unobservable modes, namely the eigenvalues of $A_{\bar{c}o}, A_{c\bar{o}}$, and $A_{\bar{c}\bar{o}}$, are not affected by the choices of A_k, B_k, or C_k. □

This result shows that when a controller link is added to the system such that it stabilizes all the modes that it can control and observe, it cannot destabilize other modes of the system that are already stable. Now, the following lemma gives a necessary and sufficient condition for the existence of the controller with transfer function structure K^{bin}.

Lemma 4. *There exists a controller with pattern K^{bin} that stabilizes a plant G if and only if every unstable mode of G is controllable and observable from at least one link k_{ij}, $k_{ij}^{bin} = 1$.*

Proof. From Lemma 3, we know that a link in the feedback controller cannot affect the uncontrollable or unobservable modes. Hence, any controller that stabilizes a given G must

have links such that all the unstable modes are both controllable and observable from at least one of the controller link. Also, if every unstable mode is controllable and observable from some controller links, these links can stabilize the plant. □

Lemmata 3 and 4 allow us to add links in P, since adding a link in P cannot change the controllability/observability of the plant for the other links in P. However, adding these links might cause the links in Q to lose controllability or observability of some of the modes, because links in Q are added on top of the links in P. Also, the links in Q themselves can create controllability/observability issues for subsequent links in Q.

Loss of observability/controllability can happen for two reasons: structurally or by exact cancellations. If it happens because of structural reasons, the system stays uncontrollable/unobservable for any choice of P or Q as long as it has the same structure. However, if the problem occurs because of exact cancellations, we can avoid these issues by a proper choice of the transfer function. Lemma 5 provides a methodology to design P and Q such that these cancellations are prevented. We will use the following result from [4] to prove the lemma.

Theorem 4. *Let G, H be proper rational transfer function matrices and suppose that $det[I + G(\infty)H(\infty)] \neq 0$. Then all the poles of the transfer function matrix*

$$W = \begin{bmatrix} (I+HG)^{-1} & -H(I+GH)^{-1} \\ G(I+HG)^{-1} & (I+GH)^{-1} \end{bmatrix}$$

are stable if and only if

- *GH has no unstable pole-zero cancellation, and*
- *all the poles of $(I + GH)^{-1}$ are stable.*

Proof. See [4] Theorem 5. □

Lemma 5. *Loss of controllability/observability can be prevented from each link in Q if pole-zero cancellations are avoided in PG and QS. Here, S is the closed loop transfer function that Q observes and controls.*

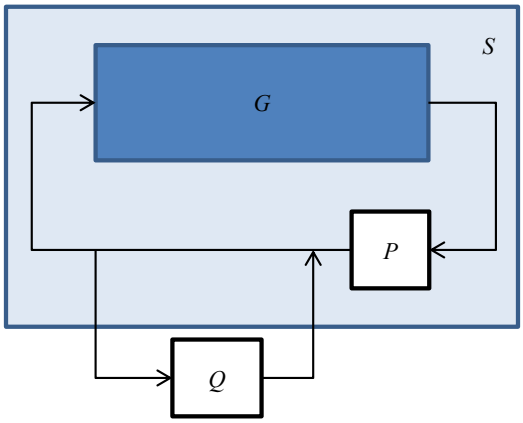


Figure 4.4: After designing P, the plant as seen by Q is given by $S = (I - PG)^{-1}$.

Proof. The transfer function that Q observes for the closed loop system formed by adding P in feedback with G is given by $S = (I - PG)^{-1}$ as shown in Figure 4.4. Using the Theorem 4, since there is no pole zero cancellations in PG, the closed loop system is stable if and only if S is stable. Which says that this transfer function has all the poles of the system. Hence Q observes and controls all the poles of the system after adding all the links in P if there is no pole zero cancellation in PG.

Similarly, when adding the links in Q if there is no pole zero cancellation in QS the controllability and observability properties are maintained. That is, if a mode is observable/controllable from a link Q_{ij} for some choices of the other links in the controller, then choosing the links in this fashion will keep the mode observable/controllable from Q_{ij}. □

Now we will present the proof of Theorem 1:

Proof. For every controller link that is added, either in P or Q, it stabilizes all the modes that are controllable and observable. Also, by Lemma 3, a newly added link cannot destabilize a

mode that was already stable. Hence with every new link added to the system, the number of unstable modes either decreases or stays the same.

If every unstable mode in the system is controllable and observable by some link, it gets stabilized. If the plant has an unstable mode that is uncontrollable or unobservable from every link in P and Q, then by Lemma 4, there is no controller with the given pattern that stabilizes the plant. Also, since the added links satisfy the conditions in Lemma 5, if a mode is controllable/observable from a link for some choice of the previously added links, then it stays controllable/observable. □

4.3 Specific Examples

In this section we use Procedure $\mathbb{P}$ to identify plants that are stabilizable or not stabilizable by controllers with some specific structural constraints.

4.3.1 Controllers with a cyclic structure

A cycle in the controller can be represented by the following binary constraints:

$$P_{cyl}^{bin} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \vdots \\ 0 & 0 & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix}_{n \times n} \quad \text{and ,}$$

$$Q_{cyl}^{bin} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \vdots \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix}_{n \times n} .$$

For such constraints on the controller we can prove the following result.

Corollary 5. *If an $n \times n$ plant is detectable and stabilizable, there always exists a stabilizing controller with the structure $(Q_{cyl}^{bin}, P_{cyl}^{bin})$.*

Proof. When all the links in P, and all but the last one in Q is added, all the remaining unstable modes of the system must be observable and controllable from the last link in Q. This happens because when adding links in the controller we satisfy the conditions in Lemma 5 avoiding any pole zero cancellations. Hence, if a link $Q_{i+1,i}$ is added then all the modes that are observable at y_i are also observable at y_{i+1}, and all the modes that are controllable from u_{i+1} are also controllable from u_i. □

4.3.2 Systems that are not stabilizable by a diagonal controller

We know that not all plants can be stabilized by a diagonal controller. To study these systems one might want to generate plants that fall in this category. We can use our results to design such systems.

From Lemma 4, we know that a detectable and stabilizable plant can be stabilized by a diagonal controller if and only if a mode of the system that is controllable from input i is also observable at the output i. Hence, a plant cannot be stabilized by a diagonal controller if there is a node that is observable only at output i and controllable only from input j, $i \neq j$. For example, the following system cannot be stabilized by a diagonal controller:

$$\dot{x} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 3 \\ 1 & 0 & 3 \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} x$$

This system has the modes at {1,2,3}. Using the Popov-Belevitch-Hautus (PBH) tests for controllability and observability, we can see that the mode 3 is controllable only from input u_1 and observable only at output y_2. Hence a diagonal controller cannot satisfy the condition given in Lemma 4.

Chapter 5

Conclusion and Future Work

This thesis explored the role of structure in the design of systems. It highlighted the fact that designing just the dynamics while ignoring the structure can make the system vulnerable. Then we provided a method to design structured controllers.

In Chapter 2, we developed some background on the dynamical structure function representation of LTI systems. We showed that this representation provides a bridge between all the other representation. We also showed that this representation has some problems that need to be resolved in the future. Such problems include, a characterization of minimal dynamical structure functions, and an efficient algorithm for finding their structural minimal realizations.

Chapter 3, explored the notion of a vulnerable link in a network of controlled linear dynamical systems. Vunerability was then defined as inverse of the size of the smallest destabilizing perturbation acting on a single link. The main results of this chapter provided necessary and sufficient conditions for the vulnerability of a link and then demonstrated that any transfer function has a completely secure architecture. This result highlights the idea that while robustness is a property of a system's dynamics, security (in the sense discussed here) is a property of its signal architecture. Future work in this area will focus on the design of low vulnerability feedback controllers for situations when the feedback cannot be avoided.

In Chapter 4, we presented an algorithm to construct stabilizing controllers with a given signal structure. We also showed that if the procedure fails to produce a stabilizing controller, the plant cannot be stabilized with a controller with the given structure. We note

that this procedure might not be a practical method for generating stabilizing controllers. This method does not provide any optimality guarantees. Also, if synthesis techniques like LQG is used to construct the controller links, the order of the transfer function on these links grows exponentially. Hence, we need to develop a controller synthesis technique that produces a low order controller and guarantees some kind of optimality. The optimality can be defined with respect to robustness, performance or other metric. These issues need to be addressed in the future research.

References

[1] S. Akhshabi and C. Dovrolis. The evolution of layered protocol stacks leads to an hourglass-shaped architecture. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 206–217, August 2011.

[2] D. L. Alderson and J. C. Doyle. Contrasting views of complexity and their implications for network-centric infrastructure. *IEEE Transactions on System, Man, and Cybernetics*, 40(4):839–852, July 2010.

[3] S. Amin, A. Cardenas, and S. S. Sastry. Safe and secure networked control systems under denial-of-service attacks. In *Proceedings of the 12th International Conference on Hybrid Systems: Computation and Control*, pages 31–45, April 2009.

[4] B. Anderson and M. Gevers. On multivariable pole-zero cancellations and the stability of feedback systems. *IEEE Transactions on Circuits and Systems*, 28(8):830–833, August 1981.

[5] A. L. Barabasi. The architecture of complexity. *IEEE Control Systems Magazine*, 27(4):33–42, August 2007.

[6] E. Byres and J. Lowe. The myths and facts behind cyber security risks for industrial control systems. In *Proceedings of the VDE Kongress*, 2004.

[7] A. A. Cardenas, S. Amin, and S. Sastry. Research challenges for the security of control systems. In *Proceedings of the 3rd Conference on Hot Topics in Security*, pages 6:1–6:6, Berkeley, CA, USA, 2008.

[8] F. A. Chandra, G. Buzi, and J. C. Doyle. Glycolytic oscillations and limits on robust efficiency. *Science*, 333(6039):187–192, 2011.

[9] J.P. Corfmat and A.S. Morse. Decentralized control of linear multivariable systems. *Automatica*, 12(5):479–495, September 1976.

[10] J. Doyle and M. Csete. Architecture, constraints, and behavior. *Proceedings of the National Academy of Science of the United States of America*, 108(3):15624–15630, July 2011.

[11] G. E. Dullerud and F. Paganini. *A Course in Robust Control Theory, a Convex Approach*. Springer, 2000.

[12] J. Goncalves, R. Howes, and S. Warnick. Dynamical structure functions for the reverse engineering of LTI networks. In *Proceedings of the IEEE/INFORMS Conference on Decision and Control*, pages 1516–1522, New Orleans, LA, December 2007.

[13] V. Hudson, P. Schrodt, and R. Whitmer. A new kind of social science: Moving ahead with reverse Wolfram models applied to event data. In *Proceedings of the 46th annual International Studies Association Convention*, Honolulu, Hawaii, March 2005.

[14] H. Ito, H. Ohmori, and A. Sano. Robust performance of decentralized control systems by expanding sequential design. *International Journal of Control*, 61(6):1297–1311, 1995.

[15] G. E. Krasner and S. T. Pope. A description of the model-view-controller user interface paradigm in the Smalltalk-80 system. Technical report, ParcPlace Systems, Inc., Mountain View, CA. `http://cincomemea.cincom.com/common/pdf/MVC_K&P.pdf`. Retrieved 6/6/2012.

[16] L. Lessard and S. Lall. Quadratic invariance is necessary and sufficient for convexity. In *Proceedings of the American Control Conference*, pages 5360–5362, 2011.

[17] Y. Liu, P. Ning, and M. Reiter. False data injection attacks against state estimation in electric power grids. *ACM Transactions on Information and System Security*, 14(1):13:1–13:33, May 2011.

[18] M. Long, C.H. Wu, and J. Y. Hung. Denial of service attacks on network-based control systems: Impact and mitigation. *IEEE Transactions on Industrial Informatics*, 1(2):85–96, May 2005.

[19] Y. Mo, E. Garone, A. Casavola, and B. Sinopoli. False data injection attacks against state estimation in wireless sensor networks. In *Proceedings of IEEE Conference on Decision and Control*, pages 5967–5972, Atlanta, GA, USA, 2010.

[20] V. Pichai, M.E. Sezer, and D.D. Siljak. Vulnerability of dynamic systems. *Proceedings of the IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, pages 409–413, December 1980.

[21] S. Ratnasamy, S. Shenker, and S. McCanne. Towards an evolvable internet architecture. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 313–324, 2005.

[22] M. Rotkowitz and S. Lall. A characterization of convex problems in decentralized control. *IEEE Transactions on Automatic Control*, 51(2):274–286, February 2006.

[23] S. Sabau and N. C. Martins. Necessary and sufficient conditions for stabilizability subject to quadratic invariance. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2459–2466, Orlando, FL, USA, December 2011.

[24] H. Sandberg, A. Teixeira, and K. H. Johansson. On security indices for state estimators in power networks. In *Proceedings of the 1st Workshop on Secure Control Systems*, 2010.

[25] D. D. Siljak. *Decentralized Control of Complex Systems*. Academic Press, Inc., 1990.

[26] A. Vasmi and N. Elia. Design of distributed controllers realizable over arbitrary directed networks. In *Proceedings of the 49th IEEE Conference of Decision and Control*, pages 4795–4800, Atlanta, GA, USA, December 2010.

[27] S. Wang and E. J. Davison. On the stabilization of decentralized control systems. *IEEE Transactions on Automatic Control*, 18(5):473–478, October 1973.

[28] Wikipedia. Stuxnet. `http://en.wikipedia.org/wiki/Stuxnet`. Retrieved 3/17/2011.

[29] E. Yeung, J. Goncalves, H. Sandberg, and S. Warnick. Representing structure in linear interconnected dynamical systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 6010–6015, December 2010.

[30] E. Yeung, J. Goncalves, H. Sandberg, and S. Warnick. Mathematical relationships between representations of structure in linear interconnected dynamical systems. In *Proceedings of the American Control Conference*, pages 4348–4353, June 2011.

[31] Y. Yuan, G. Stan, S. Warnick, and J. Goncalves. Minimal dynamical structure realisations with application to network reconstruction from data. In *Proceedings of the IEEE Conference on Decision and Control*, pages 4808–4813, 2011.