ABSTRACT

NUDELL, THOMAS R. Graph-Theoretic Algorithms for Monitoring and Control of Large Networked Dynamic Systems. (Under the direction of Aranya Chakrabortty.)

This thesis develops algorithms for monitoring and control of large networked dynamic systems with applications to large-scale power systems. Each monitoring and control deserve their own part, and while treated distinctly, they are united by a common graph-theoretic framework, namely exploiting and controlling the graph-theoretic discrete nodal domains of the underlying network.

In Part I, we develop novel measurement-based, graph-theoretic algorithms for localizing disturbances (attacks) in large clustered consensus networks. We initially develop a localization algorithm, which shows how the discrete nodal domains play a role in the measured system dynamics, for smaller graphs. We show how this algorithm can be used for localization in large clustered networks, utilizing the multi-time-scale properties of such networks. We also rigorously extend these results to second-order synchronization networks, such as swing models of power systems. Finally, using techniques for real-time topology identification, we develop an architecture in which our localization algorithm can be deployed for power system monitoring in real-time without relying on a-priori assumptions of the underlying network.

The focus of Part II shifts to developing and implementing graph-theoretic wide-area control designs to large clustered networks. Based on the results from Part I, we first design optimal state feedback control laws to ensure localizability of disturbances in the aggregate network. We develop these results for weighted tree graphs. Finally, we show how these aggregate control designs can be implemented in the full network using wide-area control inversion.

Graph-Theoretic Algorithms for Monitoring and Control of Large Networked Dynamic Systems

by
Thomas R. Nudell

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Electrical Engineering

Raleigh, North Carolina

2015

APPROVED BY:

Mo-Yuen Chow	Edgar Lobaton
Fen Wu	Aranya Chakrabortty Chair of Advisory Committee

DEDICATION

To my parents, grandparents, and brothers.

BIOGRAPHY

Tom grew up in a house in the woods outside the tiny village of South Woodbury, VT, completely off the electrical grid. His father was a cat and his mother a wolf (pronounced *woof*). He would often play in the woods with his two half-cat half-wolf brothers. Eventually, Tom got older and ran off to Kalamazoo. I know this sounds made up, but I promise it's all true. It was in Kalamazoo that Tom was first exposed to the dark arts of Mathematics and Physical Sciences. Fascinated, but delusional, he eventually hitchhiked to Raleigh (you know, like the guy in that song where they sing "thumbin' my way into North Caroline"). It was there that he began studying the dark arts more rigorously, culminating with the manuscript you read presently.

ACKNOWLEDGEMENTS

After reading my first PhD dissertation over the course of research, I knew that these few paragraphs would be the hardest to write—not because of a lack of appreciation, but because of my difficulty articulating such deep and complex gratitude for the innumerable people who have helped in ways big and small.

I would like to thank my family for their emotional, spiritual (and sometimes financial) support. I would like to thank my boubie for her wisdom and recipes which have made me famous among some circles. I would like to thank my mother for her patience and for her incredible care packages filled with everything from cheese to chocolate, which I did not start receiving until graduate school. I would like to thank my father, the English major, for editing my first papers. I would like to thank my brothers, both of who have accompanied me on this journey sometimes in body and always in spirit.

I would like to thank the FREEDM Systems Center leadership and community, and the numerous friends I have made here. I am obligated to thank the beautiful baristas of Cafe Helios—Tessa, Julia, Rhea, Lauren, Klay, and Zack—for the gallons of (free) coffee they supplied me over my five years in Raleigh; I couldn't have done this without you.

I would like to thank Lorena Bociu for reigniting my passion for applied mathematics with her course on Functional Analysis. On that note, I also need to acknowledge professors Michele Intermont and Eric Barth of Kalamazoo College for helping me realize that I had a passion for mathematics to begin with.

I would like to thank my committee, Mo-Yuen Chow, Fen Wu, and Edgar Lobaton for their insightful comments on my research. I would like to thank Edgar, in particular, whose calmness of spirit and clarity of thought seem to be contagious; I wish I could have spent more time in your office.

Finally, I cannot thank my advisor enough. Aranya has tirelessly supported and guided my academic career over the last three and a half years. He has been here with me throughout this research: from day-long whiteboard sessions to the early-morning hours before submission deadlines. Most importantly, he has been there without me—constantly advocating for our work among his colleagues everywhere he goes. For this, I am greatly beholden. Thank you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Large-scale network dynamic systems (NDS) play an ubiquitous role in the study of numerous physical and engineering systems such as robotics [BA98; Zav11; Des01], computer networks [TB86], sensor networks [Xia05], unmanned vehicles [FM04; DD03], biological and social networks [WS98] and electric power systems [Cho95], to name a few. Over the past decade tremendous research devoted to these studies has developed the theoretical foundations for modeling, monitoring, and control of typical linear and nonlinear NDS [Jad03; OS07; Zel09; ME10]. The study of NDS has been characterized as the intersection of three broad mathematical subjects [Zel09]: optimization, dynamic systems and control, and graph theory. In order to fully understand the dynamics and associated control designs for making NDS function in a desired way, the first and foremost task is to understand the graph-theory behind it.

Graph theory is a mature field of mathematics which studies the relationship between pairs of objects from a given set [Bol98]. Graph theory is closely related to other fields of mathematics including combinatorics, algebra, and geometry, while deep connections between graph theory and other branches of mathematics are routinely discovered [Mac14]. The main application of graph theory in the study of dynamic systems comes from the algebraic representations of a graph which are often found (or can be incorporated in, depending on your perspective) the state matrices of systems of networked agents. The most straightforward example of this can be found in the linear time-invariant (LTI) consensus protocol [CM13; Jad03; OS06; OS07], where the networked agents attempt to reach agreement on an objective of interest. In the most simple case, the system's state matrix *is* the graph Laplacian. This of course raises questions regarding how the underlying network topology affects the performance of the dynamic system. For consensus networks, the system's performance is closely related to the spectrum, or eigenvalues, of the graph Laplacian matrix [ME10].

The converse question was raised by Kac [Kac66] when he asked *can one hear the shape of a drum?* In terms of graph theory, the question can be summarized as *can we identify a graph*

from its Laplacian spectrum? We have known for some time that the answer is no: there are cospectral non-isomorphic graphs [Gor92] (and cospectral non-isometric graphs, in the case of weighted graphs). Recently, a better understanding of the Laplacian eigenvectors and their *nodal domains* [Biy07] has provided another way to 'listen' to graphs. And nodal domains have been proposed as a way to resolve cospectral graphs [Ban06]. Given the intrinsic relationship between the eigenvectors of a dynamic system's state matrix and the input-output transfer function, it is only natural to incorporate the nodal domain information in our study of NDS.

Another recent challenge of applied graph theory is how to deal with the explosion in network size. This growth has come hand-in-hand with an increasingly connected information-driven world. Despite ever increasing computational power, the tractability of many graph problems (some of which are NP-complete) has not kept up. One solution to this problem is to first apply a graph reduction algorithm, whereby the original graph is mapped to a smaller one that preserves (or approximates) certain properties of the original graph, e.g. its spectrum. Examples of graph reduction techniques include aggregation [CK85; Xu14] and edge sparsification [SS11].

Part of the recent growth of network sizes has been propelled by inexpensive sensors equipped with communication capabilities. These sensor networks, in general, are a major driver of the study of cyber-physical systems (CPS) [Lee08], which aims to understand the relationship between the computational (cyber) elements of the system and the physical elements. Examples of CPS include typical NDS such as sensor networks, robotics and unmanned vehicles, and power system networks. The development of advanced monitoring systems in power systems have made them a particularly interesting CPS, where the cyber network has been superimposed onto an existing physical network. This is somewhat unlike a fleet of unmanned vehicles, which without a communication network would be completely disconnected (unless two crash into one another).

An important challenge for operating any typical NDS is to ensure its security against malicious disturbances that may corrupt its dynamic performance or even lead to instability. A recent example is the STUXnet attack [FR11], which demonstrated how adversaries can exploit vulnerabilities in the cyber layers of a CPS to cause damage to the physical system. This emphasizes the pressing need for network operators to not just detect the presence of an attack or an external disturbance, but also to rapidly determine its location in the network so that appropriate control actions can be taken. We refer to this process of determining the location of a disturbance input in a large NDS as *disturbance localization*. Moreover, an equally important challenge is for the network operator to secure his or her network from vulnerabilities and potential threats.

The work presented in this thesis digs deeper into the rich and exciting field of graph theory, relates these topics to the networked system dynamics, formulates optimal control problems while considering these graph-theoretic constraints, and develops measurement-based algorithms

that are demonstrated on power system networks. Both the monitoring algorithms and control designs exploit the relationship between a reduced-order graph and the more complex network. For example, we show how the control designs can be mapped back to the more complex system, while still achieving the desired response. The results developed here are a novel set of tools which network operators can use to monitor and secure their large-scale NDS.

1.1 Contributions

The first contribution of this research is a generalization of the classic time-scale separation properties of clustered networks [CK85] to networks defined over node- and edge-weighted graphs.

In Part I we describe the relationship between the nodal domains of a graph and the impulse response transfer function residues of a consensus network defined over this graph. We show that the signs of these residues can be used to build a mapping between the input and output location, i.e. given this mapping and the output location, one can infer the input location (up to a certain level of granularity). Assuming the effect of the disturbance to be impulsive, we exploit this relationship to develop a disturbance localization algorithm in first-order consensus networks. One striking result is the characterization of the observability/controllability of a particular mode in terms of the nodal domains corresponding to that mode. We extend the basic results to synchronization networks, which are a second-order analog to consensus networks and are a mathematical representation for the swing model of a power system network. We also relax the assumption that the effect of the disturbance is impulsive. Finally, we develop a real-time implementation of our localization algorithm.

In Part II, we formulate a security-focused optimal control problem to ensure localizability of disturbances in the closed-loop aggregate network by restricting the nodal domains of each of the slow modes to come from a desired set. In other words, we are shaping, or designing, the eigenvectors of the NDS through closed-loop state feedback control while extremizing a given control objective. We next show how these optimal control designs, which are done with the aggregate network in mind, can actually be implemented in the full network through wide-area control inversion.

1.2 Organization

The topics in this thesis can be nicely grouped in two parts: Part I on monitoring and Part II on control of networked dynamic systems. Both parts take a graph-theoretic approach, utilizing the time-scale separation properties of clustered networks to efficiently deal with large-scale

networks. Additionally, the results (algorithms) throughout this theses illustrated with power system network examples.

The next chapter introduces the notations and provides fundamental mathematical results from matrix analysis, graph theory, and presents the basic model of our NDS.

Part I develops novel measurement-based graph-theoretic algorithms for localizing disturbances in network dynamic systems based on the nodal domains of the underlying graph Laplacian matrix. Chapter 3 introduces this concept in linear first-order consensus networks by relating the aforementioned nodal domains to the system's transfer function residues. The initial algorithm assumes that the disturbance can be modeled as an impulse input. Chapter 4 extends the theory to systems where each node is described by second-order equations—also called second-order synchronization networks—and to cases with a forcing function as an input. Chapter 4 also presents numerical examples to illustrate the various steps of the algorithms. Finally, Part I concludes with Chapter 5, where we develop a real-time localization algorithm that can be implemented in power system monitoring without relying on any a-priori analysis of the network.

The focus of Part II shifts to developing and implementing graph-theoretic wide-area control designs to large clustered networks. Based on the results from Part I, in Chapter 6 we first design optimal state feedback control laws to ensure localizability of disturbances in the aggregate network. We do this by considering a differential feedback control, which ensures that the closed-loop system takes the form of a graph, then we impose constraints on the nodal domains of this closed-loop graph. We implement this problem the λ_2 nodal domains of weighted trees. Finally, in Chapter 7 we show how these control designs for the aggregate network can be implemented in the full network using wide-area control inversion. We rigorously develop a decoupled multi-model composite control strategy.

Chapter 8 concludes this thesis with a summary of open problems and possible extensions to the work presented here.

Chapter 2

Mathematical Preliminaries

This chapter introduces the notation conventions, provides important graph-theoretic definitions, and reviews the standard mathematical results used throughout this thesis.

For the real or complex scalar a, we let $|a|$ denote its modulus, and for the set $\mathcal{S}$, we let $|\mathcal{S}|$ denote its cardinality. For the matrix A or the vector x, we let $\|A\|$ or $\|x\|$ denote the norm, and let $[A]_{ij}$ or $[x]_i$ denote the $(i,j)^{th}$ element or i^{th} component, respectively. We may also define a matrix with A_{ij} elements by $[A_{ij}]$, and define a diagonal matrix by $\text{diag}[x_1, \ldots, x_n]$. We let $I_n \in \mathbb{R}^{n\times n}$, $0_{n\times m} \in \mathbb{R}^{n\times m}$, and $\mathbf{1}_n \in \mathbb{R}^n$ denote the identity matrix, zeros matrix, and vector of all ones, respectively. If $0_{n\times m}$ is square, we will use the shorthand $0_n \in \mathbb{R}^{n\times n}$. We define $\mathbf{J}_n \triangleq \mathbf{1}_n\mathbf{1}_n^T$. We let $\mathbf{e}_l \in \mathbb{R}^n$ denote the unit indicator vector, where $[\mathbf{e}_l]_k = 1$ when $k = l$ and $[\mathbf{e}_l]_k = 0$ otherwise. We also define $j \triangleq \sqrt{-1}$.

2.1 Matrix Analysis

2.1.1 Eigenvalues and Eigenvectors

The eigenvalues of a square matrix $A \in \mathbb{R}^{n\times n}$ are denoted $\lambda(A) = \{\lambda_1, \ldots, \lambda_n\}$, where, by convention, we take $|\lambda_1| \leq |\lambda_2| \leq \ldots \leq |\lambda_n|$. We let v_k and w_k denote the right and left eigenvectors of A corresponding to eigenvalue λ_k. Hence, the right eigenvalue problem is written as

$$Av_i = \lambda_i v_i, \tag{2.1}$$

and the left eigenvalue problem is written as

$$A^T w_i = \lambda_i w_i. \tag{2.2}$$

We will also make use of the generalized eigenvalue problem (GEP) for the matrix pair (A, B), where $A, B \in \mathbb{R}^{n\times n}$, defined as

$$Av_i = \lambda_i B v_i. \tag{2.3}$$

The spectral radius of A is defined as

$$\rho(A) \triangleq \max_i |\lambda_i(A)| = |\lambda_n| \tag{2.4}$$

The square matrix $A \in \mathbb{R}^{n\times n}$ is said to be *positive definite* (*positive semi-definite*), denoted as $A \succ 0_n$ ($A \succeq 0_n$), if all of its eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$ are greater than zero (greater than or equal to zero).

2.1.2 Kronecker Product

We will frequently use the *Kronecker product* for matrices, which is denoted by $\otimes$. For $A = [a_{ij}] \in \mathbb{R}^{n\times m}$ and $B \in \mathbb{R}^{p\times q}$, their Kronecker product is

$$A \otimes B \triangleq \begin{bmatrix} a_{11}B & \cdots & a_{1m}B \\ \vdots & \ddots & \vdots \\ a_{n1}B & \cdots & a_{nm}B \end{bmatrix} \in \mathbb{R}^{np\times mq}. \tag{2.5}$$

It is easy to see that the Kronecker product is associative and biliear (linear in A and in B) [HJ91]. Another useful relationship is the mixed-product property: if $A \in \mathbb{R}^{n\times m},\ B \in \mathbb{R}^{p\times q},\ C \in \mathbb{R}^{m\times r},\ D \in \mathbb{R}^{q\times s}$, then

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD). \tag{2.6}$$

As a consequence of the mixed-product property, $A \otimes B$ is invertible if and only if A and B is invertible, where the inverse is

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}. \tag{2.7}$$

2.2 Graph Theory

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a collection of vertices (or nodes) $\mathcal{V} = \{1, \ldots, n\}$ and edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. The number of vertices in in $\mathcal{G}$ is $|\mathcal{V}| = n$. If there exists edge $kl \in \mathcal{E}$ then we say the nodes $k,\ l \in \mathcal{V}$ are adjacent, and write $k \sim l$. The set of nodes adjacent to $l \in \mathcal{V}$ is called the *neighborhood of* l and denoted $\mathcal{N}_l = \{k \in \mathcal{V} \mid\ k \sim l\}$. We always assume $\mathcal{G}$ is undirected, i.e., $l \sim k \Rightarrow k \sim l$, and

that there are no loops or multiple edges between nodes. We assume every node $k \in \mathcal{V}$ has a real-valued weight $m_k > 0$, and every edge $kl \in \mathcal{E}$ has a real-valued weight. Depending on the context we will denote this edge weight by $a_{kl} = a_{lk} > 0,\ kl = lk \in \mathcal{E}$ or $w_e > 0,\ e \in \mathcal{E}$. In the power system models, the node weight corresponds to the generator inertia constant, and the edge-weight is proportional to tie-line admittance. The symmetric, positive semidefinite graph Laplacian matrix associated with $\mathcal{G}$ is defined as $[L_\mathcal{G}]_{kl} = -a_{kl}$ when $k \sim l$, $[L_\mathcal{G}]_{kl} = \sum_{i \in \mathcal{N}_k} a_{ki}$ when $k = l$, and $[L_\mathcal{G}]_{kl} = 0$ otherwise. The graph Laplacian matrix can be written as

$$L_\mathcal{G} = D_\mathcal{G} - A_\mathcal{G}, \tag{2.8}$$

where the *adjaceny matrix* $[A_\mathcal{G}]_{ij} = a_{ij}$ if $ij \in \mathcal{E}$ and $[A_\mathcal{G}]_{ij} = 0$ otherwise and the diagonal *degree matrix* $[D_\mathcal{G}]_{ii} = \sum_{k \in \mathcal{N}_i}^{n} a_{ik}$. The form (2.8) is a definition of the graph Laplacian from a node-based perspective. We can also define an edge-based perspective as

$$L_\mathcal{G} = E_\mathcal{G} W E_\mathcal{G}, \tag{2.9}$$

where $W = \mathrm{diag}[w_{e_1}, \ldots, w_{e_m}],\ \forall e_i \in \mathcal{E}$ is the edge weight matrix, and the (oriented) incidence matrix $[E_\mathcal{G}]_{ik} = 1$ if edge $e_k \in \mathcal{E}$ is incident *from* node $i \in \mathcal{V}$, $[E_\mathcal{G}]_{jk} = -1$ if edge e_k is incident *to* node j, $[E_\mathcal{G}]_{ik} = 0$ otherwise. Notice that in undirected graphs the orientation is arbitrary, so by convention, if $ij = e_k \in \mathcal{E}$ with $i < j$ we write $[E_\mathcal{G}]_{ik} = 1$ and $[E_\mathcal{G}]_{jk} = -1$. An example of both definitions of the graph Laplacian is shown in Fig. 2.1.

It will be usefull to define the space of Laplacian matrices for a graph $\mathcal{G}$ as

$$\mathcal{L}_\mathcal{G} \triangleq \{L_\mathcal{G} = E_\mathcal{G} W E_\mathcal{G} \mid\ W \succ 0_n\} \tag{2.10}$$

Unless specifically noted, we assume $\mathcal{G}$ is connected, and the eigenvalues of $L_\mathcal{G}$ are ordered as $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n$. When there is no ambiguity we will drop the explicit reference to $\mathcal{G}$. We also define the *asymmetric Laplacian* matrix $L_m \triangleq M^{-1}L$, where $M = \mathrm{diag}[m_1, \ldots, m_n] \in \mathbb{R}^{n \times n}$ is the diagonal node weight matrix. Clearly L_m is also positive semidefinite.

2.2.1 Discrete Nodal Domains

We will use the following definition of weak discrete nodal domains, which we will simply refer to as *nodal domains*.

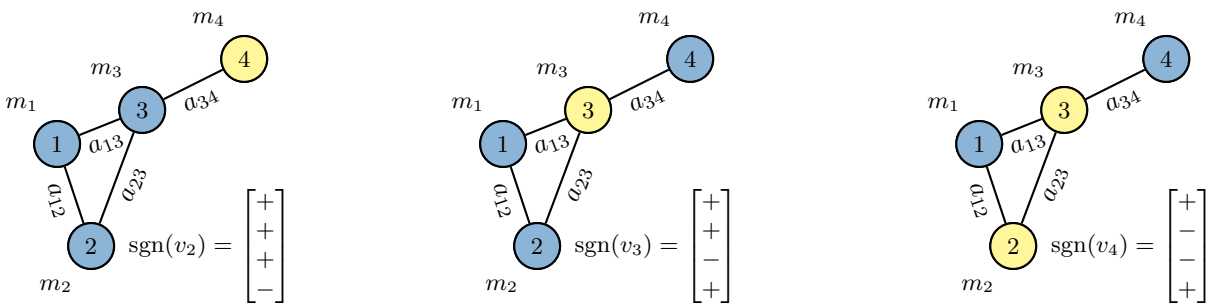
Definition 2.2.1 ((Weak) Discrete Nodal Domain [Biy07])**.** *A positive (negative) nodal domain $\mathcal{D}$ of a real-valued vector $x \in \mathbb{R}^n$ is a maximal connected subgraph of $\mathcal{G}$ on nodes $k \in \mathcal{V}$ such that $[x]_k \geq 0$ ($[x]_k \leq 0$), where $|\mathcal{V}| = n$.*

Throughout this thesis we will refer to the nodal domains of v_i (w_i) as the right (left) λ_i

$$A_{\mathcal{G}} = \begin{bmatrix} 0 & 1 & 1/2 & 0 \\ 1 & 0 & 1/4 & 0 \\ 1/2 & 1/4 & 0 & 1/5 \\ 0 & 0 & 1/5 & 0 \end{bmatrix} \qquad E_{\mathcal{G}} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & -1 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

$$M = \mathrm{diag}[1,\ 3/2,\ 1,\ 2] \qquad W = \mathrm{diag}[1,\ 1/2,\ 1/4,\ 1/5]$$

(a) **(b)**



(c) **(d)** **(e)**

Figure 2.1 Example 1: (a) The weighted adjacency matrix $A_{\mathcal{G}} = [a_{ij}]$ and diagonal node-weight matrix M. (b) The diagonal edge-weight matrix W and the incidence matrix $E_{\mathcal{G}}$. (c)–(e) The eigenvector sign patterns and nodal domains of the asymmetric Laplacian corresponding to λ_2–λ_4. (c) There are two λ_2 nodal domains $\mathcal{D}_1 = \{1, 2, 3\}$, $\mathcal{D}_2 = \{4\}$. (d) There are three λ_3 nodal domains $\mathcal{D}_1 = \{1, 2\}$, $\mathcal{D}_2 = \{3\}$, $\mathcal{D}_3 = \{4\}$. (e) There are also only three λ_4 nodal domains $\mathcal{D}_1 = \{1\}$, $\mathcal{D}_2 = \{2, 3\}$, $\mathcal{D}_3 = \{4\}$.

nodal domains, where λ_i is an eigenvalue of $L_{\mathcal{G}}$ and v_i and w_i are the corresponding right and left eigenvectors. An example of right λ_2, λ_3, and λ_4 nodal domains of a 4-node node- and edge-weighted graph is shown in Fig. 2.1. The figure illustrates the relationship between the sign pattern of the elements of the eigenvectors and the nodal domains of $\mathcal{G}$ for λ_2, λ_3, and λ_4.

Finally, from Definition 2.2.1, the following definition of *adjacent* nodal domains is obvious.

Definition 2.2.2. *Two λ_i nodal domains $\mathcal{D}_1$ and $\mathcal{D}_2$ are adjacent if there exist vertices $k \in \mathcal{D}_1$ and $l \in \mathcal{D}_2 \setminus \mathcal{D}_1$ such that the edge $kl \in \mathcal{E}$, where '$\setminus$' denotes the set-minus operation.*

2.3 Consensus Networks

The basic NDS model that we study throughout this thesis is a linear time-invariant (LTI) system defined over a connected, node- and edge-weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{V} = \{1, \ldots, n\}$,

with each node or *agent* represented by first-order consensus dynamics as

$$m_i \dot{x}_i(t) = \sum_{j \in \mathcal{N}_i} a_{ij}(x_j(t) - x_i(t)) + \sum_{j=1}^{m} b_{ij} u_j(t) + \tilde{B}_{il} \tilde{u}_l(t), \tag{2.11}$$

for $i \in \mathcal{V}$. For a given t, the state $x_i(t) \in \mathbb{R}$, the real number $m_i > 0$ represents the weight of the ith node, the real number $a_{ij} = a_{ji} > 0$ represents the weight of the edge $ij \in \mathcal{E}$, and the binary number $\tilde{B}_{il} = 1$ if the disturbance input $\tilde{u}_l(t) \in \mathbb{R}$ enters at node i, otherwise $b_{il} = 0$. The control inputs for $i \in \mathcal{V}$ are denoted by $b_{ij} u_j(t) \in \mathbb{R}$ and the locations of the disturbance inputs $\tilde{u}_l(t)$ (i.e. $\{i \in \mathcal{V} \mid \tilde{B}_{il} = 1\}$) are unknown.

The motivation for using model (2.11) is to represent the most general form of linear consensus. For example, if the time-scale of evolution of the different node-states are significantly different from each other, then the node weights $m_1, \ldots, m_n$ may be viewed as the respective time-constants associated with each node. Models similar to (2.11) have also been explored recently for optimization of various consensus problems in [Sha12a]. It also forms the basic electro-mechanical dynamics of electric power systems, which is described in detail in Appendix C. The overall system dynamics can be vectorized as

$$M\dot{x}(t) = -L_{\mathcal{G}} x(t) + Bu(t) + \tilde{B}\tilde{u}(t), \tag{2.12}$$

where $L_{\mathcal{G}} \in \mathbb{R}^{n \times n}$, $x(t) = [x_1(t), \ldots, x_n(t)]^T$, $M = \mathrm{diag}[m_1, \ldots, m_n]$, $u(t) = [u_1(t), \ldots, u_m(t)]^T$, $B \in \mathbb{R}^{n \times m}$, $\tilde{u}(t) = [\tilde{u}_1(t), \ldots, \tilde{u}_{\tilde{m}}(t)]^T$, and $\tilde{B} \in \mathbb{R}^{n \times \tilde{m}}$, with columns of $\tilde{B}$ of the form $\mathbf{e}_i$. By defining $L_m = M^{-1} L_{\mathcal{G}}$, we can write (2.12) as

$$\dot{x}(t) = -L_m x(t) + B_m u(t) + \tilde{B}_m \tilde{u}(t), \tag{2.13}$$

where $B_m = M^{-1}B$ and $M^{-1}\tilde{B}_m = \tilde{B}$. Given a set $\mathcal{S} \subset \mathcal{V}$ such that the states $\{x_j(t) | j \in \mathcal{V}\}$ of (2.13) are available from sensor measurements, the output equation can be written as

$$y(t) = Cx(t), \tag{2.14}$$

where $y(t) = [y_1(t), \ldots, y_h(t)]^T$ with $h = |\mathcal{S}|$ and the rows of $C \in \mathbb{R}^{h \times n}$ are of the form $\mathbf{e}_j^T$, $j \in \mathcal{S}$.

2.4 Time-scale Separation

In [Cho95; CK85] the authors show the two-timescale behavior of $x(t)$ in (2.13) constrained to certain conditions on the edge weights a_{ij} or on the network topology, assuming the node

weights are unity, i.e. $M = I_n$. We generalize these results for $M = \text{diag}[m_1, \ldots, m_n] \succ 0_n$.

First partition the nodes of the graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ from (2.12) into p distinct areas as $\mathcal{V} = \bigcup_{k=1}^{p} \mathcal{V}^k,\ \mathcal{V}^i \cap \mathcal{V}^j = \emptyset,\ i \neq j$, where the number of nodes in the kth area is denoted $n_k = |\mathcal{V}^k|$. The kth area may be taken as the induced subgraph $\mathcal{G}^k = \mathcal{G}[\mathcal{V}^k]$. We also define the set of inter-area edges as $\mathcal{E}^E = \{ij \in \mathcal{E}\ |i \in \mathcal{V}^i,\ j \in \mathcal{V}^j,\ i \neq j\}$, the inter-area neighborhood of any node as $\mathcal{N}_i^E = \{j \in \mathcal{V}\ |ij \in \mathcal{E}^E\}$, and the inter-area graph as $\mathcal{G}^E = (\mathcal{V}, \mathcal{E}^E)$. Following this notation, we write $\mathcal{G} = \bigcup_{k=1}^{p} \mathcal{G}^k \cup \mathcal{G}^E$. Momentarily ignoring the disturbances inputs $\tilde{B}\tilde{u}$, the dynamic network in (2.12) can then be expressed as

$$M\dot{x}(t) = -(L^I + L^E)x(t) + Bu(t), \tag{2.15}$$

where the internal Laplacian $L^I \triangleq \text{diag}[L^1, \ldots, L^p] \in \mathbb{R}^{n \times n}$ is block-diagonal with $L^k \triangleq L(\mathcal{G}^k) \in \mathbb{R}^{n_k \times n_k}$, and the external (inter-area) Laplacian is $L^E = L(\mathcal{G}^E) \in \mathbb{R}^{n \times n}$.

In the following analysis, we derive a condition on the network parameters that guarantees two time-scale behavior when explicitly considering the node weights. To understand this, we may define slow and fast variables from the states in (2.15) in the following way [CK85]. The fast variable is defined in terms of a reference node in each area, which we choose to be the n_k^{th} node in Area k, $\forall k = 1, \ldots, p$. The fast variable $x_f(t) \in \mathbb{R}^{(n-p)}$ is then defined as

$$x_f(t) = Fx(t) = \text{diag}[F^1, \ldots, F^p]x(t), \tag{2.16}$$

where F^k is a transformation for the kth area, defined as

$$F^k = \begin{bmatrix} 1 & 0 & \cdots & 0 & -1 \\ 0 & 1 & \cdots & 0 & -1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -1 \end{bmatrix}, \tag{2.17}$$

with $F^k \in \mathbb{R}^{(n_k-1) \times n_k}$. In contrast, the slow variable $x_s(t) \in \mathbb{R}^p$ represents the aggregate dynamics of each area, and is defined as

$$x_s(t) = (U^T M U)^{-1} U^T M x(t), \tag{2.18}$$

where $U = \text{diag}[\mathbf{1}_{n_1}, \ldots, \mathbf{1}_{n_p}]$. Combining the slow and fast transformations we have

$$\begin{bmatrix} x_s(t) \\ x_f(t) \end{bmatrix} = \begin{bmatrix} (U^T M U)^{-1} U^T M \\ F \end{bmatrix} x(t). \tag{2.19}$$

The inverse transformation of (2.19) is

$$x(t) = \begin{bmatrix} U & M^{-1}F^T(FM^{-1}F^T)^{-1} \end{bmatrix} \begin{bmatrix} x_s(t) \\ x_f(t) \end{bmatrix}. \tag{2.20}$$

Differentiating (2.19) with respect to t and substituting (2.15) yields

$$\begin{bmatrix} \hat{M}\dot{x}_s(t) \\ \dot{x}_f(t) \end{bmatrix} = -\begin{bmatrix} U^T \\ FM^{-1} \end{bmatrix} (L^I + L^E)x(t) + \begin{bmatrix} U^T \\ FM^{-1} \end{bmatrix} Bu(t), \tag{2.21}$$

where $\hat{M} = (U^T MU)$ is the aggregate weight matrix. Finally, substituting (2.20) into (2.21) yields

$$\begin{bmatrix} \hat{M}\dot{x}_s(t) \\ \dot{x}_f(t) \end{bmatrix} = -\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_s(t) \\ x_f(t) \end{bmatrix} + \begin{bmatrix} U^T \\ FM^{-1} \end{bmatrix} Bu(t), \tag{2.22}$$

where the state matrices are given as

$$A_{11} = U^T L^E U, \qquad A_{12} = U^T L^E M^{-1} F^T (FM^{-1}F^T)^{-1}, \tag{2.23}$$

$$A_{21} = FM^{-1}L^E U, \qquad A_{22} = FM^{-1}(L^I + L^E)M^{-1}F^T \left(FM^{-1}F^T\right)^{-1}. \tag{2.24}$$

In (2.23) we used $U^T L^I = 0$, and in (2.24) we used $L^I U = 0$. The next proposition suggests that in addition to the network sparsity and the magnitude of the edge-weights, the node-weights also play a role in clustering.

Proposition 2.4.1. *Defining the slow time-scale $t_s = t(\varepsilon \underline{d}^I)$, the system (2.22) can be written in the explicit singular perturbation form*

$$\hat{M}\dot{x}_s(t_s) = \overline{A}_{11}x_s(t_s) + \overline{A}_{12}x_f(t_s) + \overline{B}_1 u(t_s), \tag{2.25a}$$

$$\varepsilon_m \dot{x}_f(t_s) = d\overline{A}_{21}x_s(t_s) + \overline{A}_{22}x_f(t_s) + \overline{B}_2 u(t_s), \tag{2.25b}$$

with regular perturbation parameter $d = (\overline{d}^E \overline{m})/(\underline{d}^I \underline{m})$ and with the singular perturbation parameter

$$\varepsilon_m = \overline{m}\varepsilon = \overline{m}\frac{\overline{\gamma}^E}{\underline{d}^I \underline{\tilde{M}}}, \tag{2.26}$$

where

$$\underline{m} = \min_{i,k} m_i,\ i \in \mathcal{V}^k, \qquad \overline{m} = \max_{i} m_i,\ i \in \mathcal{V} \tag{2.27a}$$

$$\underline{\tilde{M}} = \min_{k} \sum_{j \in \mathcal{V}^k} m_j, \qquad \overline{\gamma}^E = \max_{k} \sum_{i \in \mathcal{V}^k,\ j \in \mathcal{N}_i^E} a_{ij},\ k = 1, \ldots, p, \tag{2.27b}$$

$$\underline{d}^I = \min_{i,\,k} \sum_{j \in \mathcal{V}^k} a_{ij} \qquad \overline{d}^E = \max_{i,\,k} \sum_{j \in \mathcal{N}_i^E} a_{ij},\ i \in \mathcal{V}^k,\ k = 1, \ldots, p. \tag{2.27c}$$

The normalized state matrices in (2.25) are defined as

$$\overline{A}_{11} = A_{11}/\overline{\gamma}^E, \qquad \overline{A}_{12} = A_{12}/\overline{\gamma}^E, \tag{2.28a}$$

$$\overline{A}_{21} = A_{21}/(\overline{d}^E/\underline{m}), \qquad \overline{A}_{22} = A_{22}/(\underline{d}^I/\overline{m}), \tag{2.28b}$$

$$\overline{B}_1 = U^T B/\overline{\gamma}^E, \qquad \overline{B}_2 = F M^{-1} B/(\underline{d}^I/\overline{m}). \tag{2.28c}$$

Proof. See Appendix A.1. □

The parameter $\overline{m}\overline{\gamma}^E$ in (2.26) is a measure of the worst-case edge sparsity between the areas, while $\underline{d}^I \underline{\tilde{M}}$ is a measure of best-case edge density inside the areas. Therefore, a direct impilication of Assumption 3.0.1 is $\overline{m}\overline{\gamma}^E \ll \underline{d}^I \underline{\tilde{M}}$ or $\varepsilon_m \ll 1$. This, in turn, using Proposition 2.4.1, means that the system (2.13) exhibits a two time-scale behavior [Cho95]. Using this result, the next three chapters relate the residues corresponding to the $p-1$ slow poles of the transfer matrix of (2.13) to the graph-theoretic properties of $\mathcal{G}$ in order to develop our localization algorithms. Again, this formulation will be extremely convenient when n is large compared to the number of areas p. For example, if $n = 1000$ but $p = 5$, then one would need to only compute the four residues associated with the eigenvalues $0 \neq \lambda_2 \leq \lambda_3 \leq \lambda_4 \leq \lambda_5$ to localize the inputs.

Part I

Monitoring

Chapter 3

Disturbance Localization in Consensus Networks

We consider the open-loop consensus network, i.e. ignoring the control inputs $Bu(t)$ in (2.13), defined over the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a chosen set $\mathcal{S} \subseteq \mathcal{V}$ of measured states as

$$\dot{x}(t) = -L_m x(t) + \tilde{B}_m \tilde{u}(t), \tag{3.1a}$$

$$y(t) = Cx(t), \tag{3.1b}$$

where $x(t) \in \mathbb{R}^n$, $L_m = M^{-1}L_{\mathcal{G}}$, and $\tilde{B} \in \mathbb{R}^{n\times\tilde{m}}$ are defined in (2.13), $\tilde{u}(t) \in \mathbb{R}^{\tilde{m}}$ are the unknown disturbance inputs, and $y(t) =\in \mathbb{R}^h$ with $h = |\mathcal{S}|$, and the rows of $C \in \mathbb{R}^{h\times n}$ are of the form $\mathbf{e}_j^T$, $j \in \mathcal{S}$. Intuitively our goal is to keep h small. We will discuss more details of how $\mathcal{S}$ should be chosen in Section 3.2.

Our objective is to localize the disturbance inputs $\tilde{u}_j(t)$, $j = 1, \ldots, \tilde{m}$ in (3.1a). This, however, cannot be done directly through system identification methods, which are only accurate up to a similarity transform of the pair $(-L_m,\ \tilde{B}_m)$. Furthermore, if n is large, then finding the precise set of nodes where the inputs enter may be computationally intractable within a given time-frame. To bypass this limitation we make the following assumption:

Assumption 3.0.1. *The graph $\mathcal{G}$ in (2.13) can be partitioned into p coherent areas [Cho95], also called clusters [GN02], where edges within clusters are dense, and edges between clusters are sparse.*

Such clustering is prevalent throughout engineered and natural networked systems including power transmission systems, social and biological networks, the internet and the world-wide web, etc. [GN02; WS98]. For example, as shown in [CK85], the U.S. West Coast power grid can be partitioned into as many as 11 coherent areas. In light of Assumption 3.0.1 we, therefore,

formulate our problem for Part I as follows:

Given $y(t)$ from (5.2), develop an algorithm to identify the cluster in which the disturbance input in (3.1a) $\tilde{u}_j(t),\ j = 1, \ldots, \tilde{m}$, enter.

The results in Section 2.4 show that Assumption 3.0.1 leads to a two time-scale behavior of $x(t)$ in (3.1a). This two time-scale behavior of $x(t)$ will indeed enable us to identify the cluster(s) in which the inputs $\tilde{u}_j(t),\ j = 1, \ldots, \tilde{m}$, may have entered. This formulation is also particularly attractive when the individual areas are managed by non-cooperative local operators, as will be shown in the examples of Section 4.4.

3.1 Relating Residues and Nodal Domains

In this section we relate the residues of the transfer function for every input-output pair of (3.1a) and (5.2) to the nodal domains of $\mathcal{G}$. We do this in two steps. We first relate the residue R_{ij}^k corresponding to the kth slow pole of the transfer function from the ith input to the jth output to the respective left and right eigenvectors of L_m. Thereafter, we exploit the algebraic relationship between these eigenvectors and the nodal domains of $\mathcal{G}$ to obtain an explicit relationship between R_{ij}^k and the nodal domains.

3.1.1 Relating Residues to Eigenvectors of L_m

The transfer matrix of (3.1a)–(5.2) can be constructed as $G(s) = C(sI_n + L_m)^{-1}B_m$. The (i, j)-th element of $G(s)$, defined as $g_{ij}(s) \triangleq [G(s)]_{ij}$, is a rational function of the form

$$g_{ij}(s) = \frac{\beta_1 s^{n-1} + \beta_2 s^{n-2} + \cdots + \beta_{n-1} s + \beta_n}{s(\alpha_2 s^{n-1} + \alpha_3 s^{n-2} + \cdots + \alpha_{n-1} s + \alpha_n)}. \tag{3.2}$$

Since the network is clustered into p areas, L_m will have $p - 1$ non-zero slow eigenvalues [AF03]. For simplicity, we assume that the slow eigenvalues are simple, in which case the transfer function in (3.2) can be decomposed into pole-residue form as

$$g_{ij}(s) = \underbrace{\sum_{k=2}^{p} \frac{R_{ij}^k}{s + \lambda_k}}_{g_{ij}^{slow}(s)} + \bar{g}_{ij}(s), \tag{3.3}$$

where $\bar{g}_{ij}(s)$ contains the zero mode and the fast modes. The residue R_{ij}^k can also be represented as the product of the mode controllability factor and mode observability factor [ML89] as

$$R_{ij}^k = [C v_k w_k^T B_m]_{ji} = \underbrace{\left(\mathbf{e}_j^T v_k\right)}_{\text{obsv. factor}} \underbrace{\left(w_k^T M^{-1} \mathbf{e}_i\right)}_{\text{ctrl. factor}}, \tag{3.4}$$

where v_k and w_k are the right and left eigenvectors, respectively, associated with the kth slow eigenvalue of L_m. We next extrapolate the relationship between R_{ij}^k and the eigenvectors v_k and w_k in (5.8) to the nodal domains of $\mathcal{G}$.

3.1.2 Relating Eigenvectors of L_m to Nodal Domains of $\mathcal{G}$

We start with three important lemmas. They establish the relationship between the nodal domains of the left and right Laplacian eigenvectors of a node- and edge-weighted graph $\mathcal{G}$ to the residues in (5.8), and they are the building blocks of the main result of this chapter.

Lemma 3.1.1. *The nodal domains of the left and right eigenvectors of L_m are identical.*

Proof. Let w_k and v_k be the left and right eigenvectors of L_m corresponding to λ_k. It suffices to show that $\text{sgn}([v_k]_i) = \text{sgn}([w_k]_i)$, $\forall i \in \mathcal{V}$. The left eigenvector must satisfy $L_m^T w_k = \lambda_k w_k$ [HJ85]. It follows that $(M^{-1}L)^T w_k = \lambda_k w_k = \lambda_k M(M^{-1}w_k)$. Pre-multiplying by M^{-1} yields $L_m(M^{-1}w_k) = \lambda_k(M^{-1}w_k)$. Therefore $v_k = \alpha_k M^{-1} w_k$, where $\alpha_k > 0$ is a real-valued constant. The assumption that $M^{-1} > 0$ implies $\text{sgn}([v_k]_i) = \text{sgn}([w_k]_i)$ $\forall i \in \mathcal{V}$. □

As a result of this relationship between the left and right eigenvectors of L_m the residues of the transfer function defined in (5.8) are symmetric, as stated in the following corollary.

Corollary 3.1.1. *The residues defined in (5.8) are symmetric, i.e., $R_{ij}^k = R_{ji}^k$, $i, j \in \mathcal{V}$, $k = 1, \ldots, n$.*

Proof. Consider the arbitrary input-output pair (i, j). The output matrix $C = \mathbf{e}_j^T$ and the input matrix $B_m = M^{-1}\mathbf{e}_i$. The residue associated with the kth eigenvalue is $R_{ij}^k = \mathbf{e}_j^T v_k w_k^T M^{-1}\mathbf{e}_i$. Next, define the matrix of residues $[R_{ij}^k]$ by considering every input-output pair,

$$[R_{ij}^k] = \left(\begin{bmatrix} \mathbf{e}_1^T \\ \hline \mathbf{e}_2^T \\ \hline \vdots \\ \hline \mathbf{e}_n^T \end{bmatrix} v_k \right) \left(w_k^T M^{-1} \left[\begin{array}{c|c|c|c} \mathbf{e}_1 & \mathbf{e}_2 & \cdots & \mathbf{e}_n \end{array} \right] \right) = v_k w_k^T M^{-1}. \tag{3.5}$$

From Lemma 3.1.1 $v_k = \alpha_k M^{-1} w_k$, and we have $[R_{ij}^k] = \frac{1}{\alpha_k} v_k v_k^T = [R_{ij}^k]^T$, which implies $R_{ij}^k = R_{ji}^k$, $\forall i, j \in \mathcal{V}$. □

At this time, we define the symmetrized Laplacian, which is used in the next two lemmas, as

$$L_s = M^{-1/2} L M^{-1/2}. \tag{3.6}$$

By construction, L_s is similar to L_m, i.e., they have the same eigenvalues [HJ85]. Moreover, since $M > 0$, the nodal domains of L_s are identical to those of L_m.

Lemma 3.1.2. *Let v_k and ψ_k denote eigenvectors L_m and L_s, respectively, corresponding to the eigenvalue λ_k. The nodal domains of v_k and ψ_k are identical.*

Proof. As in Lemma 3.1.1, it suffices to show $\text{sgn}([v_k]_i) = \text{sgn}([\psi_k]_i), \forall i \in \mathcal{V}$. By definition, the eigenvector v_k satisfies $L_m v_k = M^{-1} L v_k = \lambda_k v_k$. Pre-multiplying both sides by $M^{1/2}$ yields

$$M^{-1/2} L v_k = M^{-1/2} L M^{-1/2} (M^{1/2} v_k) \tag{3.7}$$

$$= L_s (M^{1/2} v_k) = \lambda_k (M^{1/2} v_k). \tag{3.8}$$

It follows that $\psi_k = M^{1/2} v_k$ is the eigenvector of L_s corresponding to λ_k. Since $M^{1/2} = \text{diag}[\sqrt{m_1}, \ldots, \sqrt{m_n}] > 0$, therefore $\text{sgn}([\psi_k]_i) = \text{sgn}([v_k]_i)\ \forall i \in \mathcal{V}$. □

This result guarantees that nodal domains are preserved under similarity transformations of the form (3.6), even when eigenvectors are not. Using Lemma 3.1.1 and Lemma 3.1.2, the next result relates the signs of the residues of the transfer function in (6.5) to the nodal domains of L_m.

Lemma 3.1.3. *Let v_k be an eigenvector of L_m (or its similarity transformation L_s) corresponding to λ_k, and let $\mathcal{D}_1$ and $\mathcal{D}_2$ be adjacent nodal domains of v_k. Then 1) $R_{ij}^k > 0$ if $i, j \in \mathcal{D}_1 \setminus \mathcal{D}_2$; 2) $R_{ij}^k < 0$ if $i \in \mathcal{D}_1 \setminus \mathcal{D}_2$ and $j \in \mathcal{D}_2 \setminus \mathcal{D}_1$; and 3) $R_{ij}^k = 0$ if $i \in \mathcal{D}_1 \cap \mathcal{D}_2$ or $j \in \mathcal{D}_1 \cap \mathcal{D}_2$.*

Proof. From Lemmas 3.1.1 and 3.1.2, $\text{sgn}(R_{ij}^k) = \text{sgn}([v_k]_i [v_k]_j)$. Furthermore, by definition, adjacent nodal domains have opposite signs. It follows that

1. $i, j \in \mathcal{D}_1 \setminus \mathcal{D}_2$ implies both $[v_k]_i, [v_k]_j \neq 0$ and $\text{sgn}([v_k]_i) = \text{sgn}([v_k]_j)$. Thus, $[v_k]_i [v_k]_j > 0$.
2. $i \in \mathcal{D}_1 \setminus \mathcal{D}_2$ and $j \in \mathcal{D}_2 \setminus \mathcal{D}_1$ implies $[v_k]_i \neq 0$ and $[v_k]_j \neq 0$. Since i and j are in adjacent domains, $\text{sgn}([v_k]_i) = -\text{sgn}([v_k]_j)$. Thus, $[v_k]_i [v_k]_j < 0$.
3. $i \in \mathcal{D}_1 \cap \mathcal{D}_2 \Rightarrow [v_k]_i = 0$, thus $[v_k]_i [v_k]_j = R_{ij}^k = 0$, and if $j \in \mathcal{D}_1 \cap \mathcal{D}_2$, then $R_{ij}^k = 0$. □

This suite of results will be used to construct our localization algorithm based on the signs of the residues corresponding to the slow poles of (6.5). In particular, these signs can be used to construct a so called *localization key* which maps inputs in Area i to outputs in Area j for all areas $i, j = 1, \ldots, p$. An *a-priori* step of our algorithm is to construct a $p \times p$ array containing all of these localization keys defined for a nominal model of (3.1a), as explained next.

3.1.3 Input Localization Keys

We assume there exists a centralized network supervisor who knows a baseline model for (3.1a). This supervisor, therefore, has knowledge of the nominal clustering structure of the network

graph $\mathcal{G}$, and the nominal Laplacian matrix, denoted as $\overline{L}_m$. Using $\overline{L}_m$, he constructs a $p \times p$ array of nominal localization keys, denoted as $\mathcal{K} = [\mathcal{K}_{ij}]$, where $\mathcal{K}_{ij}$ is the set of signs $\{+, -, 0\}$ of the residues corresponding to the $p-1$ slow eigenvalues of $\overline{L}_m$. For definiteness let

$$\mathcal{K}_{ij} = \{\text{sgn}(\overline{R}^2_{q_i q_j}), \ldots, \text{sgn}(\overline{R}^p_{q_i q_j})\}, \tag{3.9}$$

where $\overline{R}^k_{q_i q_j} = [\overline{v}_k \overline{w}_k^T M^{-1}]_{q_j q_i}$ from (5.8) with $q_i \in \mathcal{V}^i,\ q_j \in \mathcal{V}^j$, and $\overline{v}_k$ and $\overline{w}_k$ denote the right and left eigenvectors corresponding to the kth slow eigenvalue of $\overline{L}_m$, respectively. Construction of these keys is a one-time task as long as the clustering structure of L_m for any disturbance event remains the same as that of $\overline{L}_m$. In other words, during a disturbance, the network parameters may change from their nominal values (e.g., changes in generator inertias and tie-line reactances in power grids), or some edges may be removed (e.g., loss of tie-lines or communication links between agents), but as long as the network retains the nominal clustering structure despite these changes one may use the same array of localization keys to determine the location of the incoming inputs.

Example: An example of constructing $\mathcal{K}$ for a small four-node graph is shown in Fig. 2.1. Each node can be considered as an equivalent area, resulting in a 4-area partitioning (one may also consider a 3-area partitioning of this graph, as we do in Section 3.3). The key length is $p-1=3$. We consider the nodal domains corresponding to $\lambda_2,\ \lambda_3$, and λ_4. The sign patterns of the corresponding eigenvectors are shown in Fig. 2.1. The localization keys can be constructed using (3.9) and Lemma 3.1.3. One may also compare the color of the nodes: i.e., for a given λ_k if two nodes are of the same color, then the corresponding residue is positive, and vice-versa. For example, $\text{sgn}(R^2_{12}) = \{+\},\ \text{sgn}(R^3_{12}) = \{+\}$, and $\text{sgn}(R^4_{12}) = \{-\}$, and therefore the localization key $\mathcal{K}_{12} = \{+, +, -\}$. The complete array of input localization keys is shown in Table 3.1. It is clear from this example that $\mathcal{K}$ is symmetric.

Following Assumption 3.0.1 and the analysis in Section 2.4, we note that the array in Table 3.1 would be applicable if each node of the graph in Fig. 2.1 were a coherent cluster with multiple internal nodes. This clearly demonstrates the advantage of relating the nodal domains of $\mathcal{G}$ in (3.1a) to the signs of the residues of the slow poles. Concretely, once the $p-1$ slow eigenvalues and eigenvectors of L_m have been computed, only an additional $p-1$ multiplications are needed to compute $\mathcal{K}_{ij}$ in (3.9). Furthermore, the $p \times p$ array is symmetric and only the off-diagonal keys are unknown, thus computing the entire array only requires a total of $(p-1)\left((p^2-p)/2\right)$ multiplications. More importantly, the local area operators will only require the row (or column) from the array corresponding to their own area. This means that no area needs to reveal its internal topology to other areas, thereby preserving the privacy of the algorithm. Furthermore, each area may execute the algorithm with its local measurements only, and does not require measurements from other areas, thereby eliminating the need for inter-area data communication.

Table 3.1 Input localization keys for example system in Fig. 2.1.

Area	1	2	3	4
1	$\{+,+,+\}$	$\{+,+,-\}$	$\{+,-,-\}$	$\{-,+,+\}$
2	$\cdot$	$\{+,+,+\}$	$\{+,-,+\}$	$\{-,+,-\}$
3	$\cdot$	$\cdot$	$\{+,+,+\}$	$\{-,-,-\}$
4	$\cdot$	$\cdot$	$\cdot$	$\{+,+,+\}$

3.2 Measurement-based Localization

Up to this point we have developed a set of graph-theoretic tools analyzing the system-theoretic properties of the NDS (3.1a). This section applies these results to construct our measurement-based disturbance localization algorithm for this NDS. The *a-priori* step, as mentioned previously, is to construct the nominal array of localization keys $\mathcal{K}$ whose row $\mathcal{K}_j = [\mathcal{K}_{1j}, \ldots, \mathcal{K}_{pj}]$ is associated with the node set $\mathcal{V}^j$. We assume that every area is managed by a local operator who has been provided $\mathcal{K}_j$. Following the intrusion of the disturbance input, the local operator must now run a system identification to estimate the transfer function $g_{ij}^{slow}(s)$ using its local measurements, and estimate the slow-mode residues $R_{ij}^k,\ k = 2, \ldots p$. A necessary requirement for estimating these residues is that $g_{ij}^{slow}(s)$ must be minimum realizable. This requirement is intrinsically related to the structure of $\mathcal{G}$. We first define the *zero nodes* of the graph $\mathcal{G}$, and then state necessary and sufficient conditions which the input and output must satisfy in terms of these zero nodes in order to guarantee minimum realizability.

Definition 3.2.1 (Zero nodes)**.** *The set of zero nodes of the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ corresponding to eigenvalue λ_k is defined as $\mathcal{Z}_k = \{i \in \mathcal{V} |\, [v_k]_i = 0\}$, where v_k is a right or left eigenvector of $L_{\mathcal{G}}$ corresponding to λ_k.*

Theorem 3.2.1. *$g_{ij}^{slow}(s)$ from (6.5) is minimum realizable if and only if the input does not enter a zero node corresponding to $\lambda_k,\ k = 2, \ldots, p$, and the output is not taken at a zero node corresponding to $\lambda_k,\ k = 2, \ldots, p$.*

Proof. From Lemma 3.1.1, zero nodes of the left and right eigenvectors are identical, so the proofs of the observability and controllability conditions are analogous. We state the controllability proof here. Since the columns of $\tilde{B}_m$ are of the form $\mathbf{e}_i/m_i$, a left eigenvector, denoted w_k, is orthogonal to the columns of $\tilde{B}_m$ if and only if $w_k^T \mathbf{e}_i/m_i = [w_k]_i/m_i = 0 \Leftrightarrow [w_k]_i = 0$. The controllability condition follows from the PBH-controllability test [AM05]. □

Theorem 3.2.1 implies that a necessary condition for the disturbance input $\tilde{u}(t)$ to be detectable is that neither the input nor the output(s) are allowed to be at the zero nodes of the

network (whenever such nodes exist). We, therefore, make the following assumption to ensure the detectability of $\tilde{u}(t)$ so that it can be localized.

Assumption 3.2.1.

- *The set $\mathcal{S}$ satisfies $\mathcal{S} \nsubseteq \mathcal{Z}_k,\ \forall k = 2, \ldots, p$.*
- *Any disturbance $\tilde{u}_j(t),\ j = 1, \ldots, m$, occurs at, $j \notin \mathcal{Z}_k,\ \forall k = 2, \ldots, p$.*

Assumption 3.2.1 is valid for all practical purposes since in reality local area operators will know the identity of their zero nodes, and, hence, they may impose extra security measures to ensure that these nodes are never attacked. In order to estimate the transfer function residues from the output measurements, the central operator must run a system identification algorithm. Two such algorithms that are widely used in the power system community are the Eigensystem Realization Algorithm (ERA) [JP85] and Prony's method [Hau90]. Basic versions of these algorithms are presented in Appendix B.

3.2.1 Localization Algorithm

After the transfer function $\hat{g}_{ij}^{slow}(s)$ has been computed, the signs of the estimated residues $\hat{R}_{ij}^{k},\ k = 2, \ldots, p$, are used to build a localization key $\hat{\mathcal{K}}_{ij}$ following (3.9). Recall that the nominal keys in the $p \times p$ array $\mathcal{K}$ in (3.9) map the inputs and outputs between any pair of areas. Therefore, as long as the clustering structure of $\mathcal{G}$ in (3.1a) has not changed from the nominal conditions, the estimated $\hat{\mathcal{K}}_{ij}$ can be matched to the elements of $\mathcal{K}$ in order to determine the area in which the disturbance may have occurred. Since the local operator in Area j only has access to the jth row $\mathcal{K}_j = [\mathcal{K}_{1j}, \ldots, \mathcal{K}_{pj}]$, he tries to match the estimated $\hat{\mathcal{K}}_{ij}$ to the elements of $\mathcal{K}_j$. If the matching is successful then the input may have occurred in the corresponding area. For example, if $\hat{\mathcal{K}}_{ij} = [\mathcal{K}_j]_1$, then the input may have occurred in Area 1. Algorithm 1 summarizes the entire measurement-based localization procedure.

Algorithm 1 Output-only localization for measurement node $q_j \in \mathcal{V}^j$ and input u_κ

Require: Measurements $y_{q_j}(t_k),\ t_k = t_0,\ t_1, \ldots, t_\varsigma$, and nominal keys $\mathcal{K}_j = [\mathcal{K}_{1j}, \ldots, \mathcal{K}_{pj}]$

```
Run System Identification to estimate (Â, B̂, Ĉ) from (B.5)
Detect slow modes λ̂_k k = 2, ..., p̂, where λ̂_p̂ ≪ λ̂_p̂+1
Estimate residues R̂^k_κj, i = 1, ..., p̂,
Construct estimated localization key as K̂_κj = {sgn(R̂^2_κj), ..., sgn(R̂^p_κj)}
for i = 1 → p do
    Match the estimated key to elements of K_j
    if Match, i.e., K̂_κj = [K_j]_i then
        Input may be located in Area i, i.e. κ ∈ V^i
    end if
end for
```

We see the importance of Assumption 3.2.1 in the first step of Algorithm 1. Without this assumption we do not have a minimum realization of the triple ($\hat{A}$, $\hat{B}$, $\hat{C}$), which means that some or all of the slow modes and their corresponding residues are not available to fully construct $\hat{K}_{ij}$. If Assumption 3.2.1 is met, we can state the following result regarding Algorithm 1.

Theorem 3.2.2 (Uniqueness of Input Localization)**.** *For an arbitrary output node $q_j \in \mathcal{V}^j$, the localization resulting from Algorithm 1 is unique iff the nominal keys $\mathcal{K}_{kj} \neq \mathcal{K}_{\ell j},\ \forall k \neq \ell$.*

Proof. Necessary condition: The assumption that the localization is unique means that $\hat{\mathcal{K}}_{\kappa j} = \mathcal{K}_{ij}$ for exactly one $i = 1, \ldots, p$, which implies $\mathcal{K}_{kj} \neq \mathcal{K}_{\ell j},\ \forall k \neq \ell$. To prove sufficiency, assume $\mathcal{K}_{kj} \neq \mathcal{K}_{\ell j},\ \forall k \neq \ell$, and that the localization is not unique. Then we have $\mathcal{K}_{kj} = \hat{\mathcal{K}}_{\kappa j} = \mathcal{K}_{\ell j} \Rightarrow \mathcal{K}_{kj} = \mathcal{K}_{\ell j}$ for some $k \neq \ell$, a contradiction. □

Theorem 3.2.2 provides the necessary and sufficient conditions for unique localization using Algorithm 1, meaning there is no ambiguity as to which area the input entered. It, however, does not indicate how the dynamic network (3.1a) should be designed to uniquely localize a disturbance input. We next develop a class of graphs to bridge this gap.

3.3 p-Area Complete Graphs

In order to satisfy the sufficient conditions provided by Theorem 3.2.2, we define a class of graphs called p-area complete graphs as follows.

Definition 3.3.1. *The connected node- and edge-weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a p-area partition is p-area complete if, for all generalized eigenvalues λ_k of $(L_\mathcal{G}, M),\ k = 2, \ldots, p$:*

- *the set of zero nodes is empty;*
- *there are exactly k nodal domains;*
- *and, each area lies in exactly one nodal domain.*

The first property ensures that the first p Laplacian eigenvalues of a p-area complete graph are simple, because eigenvectors corresponding to repeated eigenvalues have at least one vanishing coordinate [Biy07]. The second and third properties together imply that if a graph is p-area complete then it is also $(p-1)$-area complete. These conditions may appear to be artifacts, but the first and third properties hold for all practical networks that exhibit strong clustering. Property 2 places some constraints on the wide-area network topology, but to the best of our knowledge we are not aware of any other less restrictive assumptions. These conditions are sufficient for unique localization using Algorithm 1, which we establish with the following theorem.

Theorem 3.3.1. *If $\mathcal{G}$ in (3.1a) is p-area complete, then Algorithm 1 guarantees unique localization of $\tilde{u}(t)$.*

Proof. Let $q_j \in \mathcal{V}^j$ be an arbitrary output node in Area j. We will use induction on the number of nodal domains k. As a basis, consider $k = 2$. The key length is 1, and since there are no zero nodes $\mathcal{K}_{j1} = \{+\}$ corresponds to an input in the same nodal domain as q_j and $\mathcal{K}_{j2} = \{-\}$ corresponds to an input in the adjacent nodal domain as q_j. Hence the localization is unique. Now assume the statement is true for all $k < p$. Let $k = p - 1$. In that case, Theorem 3.2.2 implies that keys of length $p - 2$ are unique. Moreover, from the second and third properties of p-area complete graphs, one nodal domain contains two areas, while the remaining $p - 2$ nodal domains contain exactly one area each. Now if we let $k = p$, the two previously paired areas are in adjacent nodal domains, while the adjacency pattern of the remaining areas has not changed. Hence, the $(k-1)$-length keys $\mathcal{K}_{ji},\ i = 1, \ldots, p$ are unique. □

For example, consider the network in Fig. 2.1. Given the 3-area partitioning $\mathcal{V}^1 = \{1, 2\}$, $\mathcal{V}^2 = \{3\}$, $V^3 = \{4\}$, this graph is 3-area complete. To emphasize that p-area completeness provides only sufficient conditions for unique localization, and not *necessary* conditions, consider the 4-area partitioning of the same network. There are only three nodal domains corresponding to λ_4 (Fig. 2.1(e)), but Table 3.1 still satisfies the conditions for unique localization in Theorem 3.2.2. Also, while the construction of a p-area complete graph may initially seem restrictive from Definition 3.3.1, this property, in fact, can be found in several real-world networks. For example, the widely used aggregate model of the U.S. West Coast power grid presented in [Bal04] is 5-area complete. Random proximity graphs, which have wide applications in sensor and social networks, also exhibit similar p-area completeness properties, an example being the 20-node network

presented in [OS07] that can be easily shown to be 4-area complete under minor approximations. Furthermore, we next show that a p-area complete graph with an arbitrarily large number of nodes may be constructed by taking the Cartesian product of two smaller graphs.

3.3.1 Constructing p-area Complete Graphs using Cartesian Products

The Cartesian product[1] $\mathcal{G}_1 \square \mathcal{G}_2$, of graphs $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ has vertex set $\mathcal{V} = \mathcal{V}_1 \times \mathcal{V}_2$, where vertices (i_1, i_2), $(j_1, j_2) \in \mathcal{V}$ are adjacent if and only if either $i_1 = j_1$ and $i_2 j_2 \in \mathcal{E}_2$ or $i_2 = j_2$ and $i_1 j_1 \in \mathcal{E}_1$ [ME10]. For the remainder of this section we will assume $\mathcal{G}_1$ and $\mathcal{G}_2$ are connected graphs with $|\mathcal{V}_1| = n_1$ and $|\mathcal{V}_2| = n_2$. We next present two lemmas. The first lemma provides constructive results for obtaining a new p-area complete graph by taking the product of two edge-weighted graphs. This result will be generalized for node- and edge-weighted graphs in the second lemma.

Lemma 3.3.1. *Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be edge-weighted graphs, and define $L_1 = L_{\mathcal{G}_1}$ and $L_2 = L_{\mathcal{G}_2}$. Denote the eigenvalues and eigenvectors of L_1 as $\lambda_1 \leq \cdots \leq \lambda_{n_1}$ and $v_1, \ldots, v_{n_1}$, respectively. Denote the eigenvalues and eigenvectors of L_2 as $\mu_1 \leq \cdots \leq \mu_{n_2}$ and $u_1, \ldots, u_{n_2}$, respectively. If $\mathcal{G}_1$ is p-area complete and $\mu_2 > \lambda_p$ then the graph $\mathcal{G} = \mathcal{G}_1 \square \mathcal{G}_2$ is also p-area complete.*

Proof. Let the p-area partitioning of $\mathcal{G}$ be the same as that of $\mathcal{G}_1$, but where each of the nodes in $\mathcal{G}_1$ has been replaced by a copy of the graph $\mathcal{G}_2$. The eigenvalues and corresponding eigenvectors of $L_{\mathcal{G}}$ are [ME10]

$$\lambda'_k = \lambda_i + \mu_j, \quad 1 \leq i \leq n_1, \quad 1 \leq j \leq n_2, \tag{3.10}$$

$$v'_k = v_i + u_j, \quad 1 \leq i \leq n_1, \quad 1 \leq j \leq n_2, \tag{3.11}$$

for $k = 1, \ldots, n_1 \times n_2$. Arranging the first p of these in increasing order, we have

$$\lambda'_k = \lambda_k, \qquad v'_k = v_k \otimes u_1, \tag{3.12}$$

where $\otimes$ denotes the Kronecker product. Since $\mu_1 = 0$ and $u_1 = \mathbf{1}_{n_2}$, v'_k has the same sign pattern as v_k, $k = 1, \ldots, p$, where each of the original elements in v_1 has been replaced by a vector of length n_2. Therefore, $\mathcal{G}$ is at least p-area complete. $\square$

A generalized version of Lemma 3.3.1 can be obtained by using a node-weighted graph product, defined as follows.

Definition 3.3.2 (Node-weighted Cartesian Product)**.** *Let $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ be node- and edge-weighted graphs with diagonal node-weight matrices $M_1 \in \mathbb{R}^{n_1 \times n_1}$ and $M_2 \in$*

[1] This is sometimes called the box product of graphs.

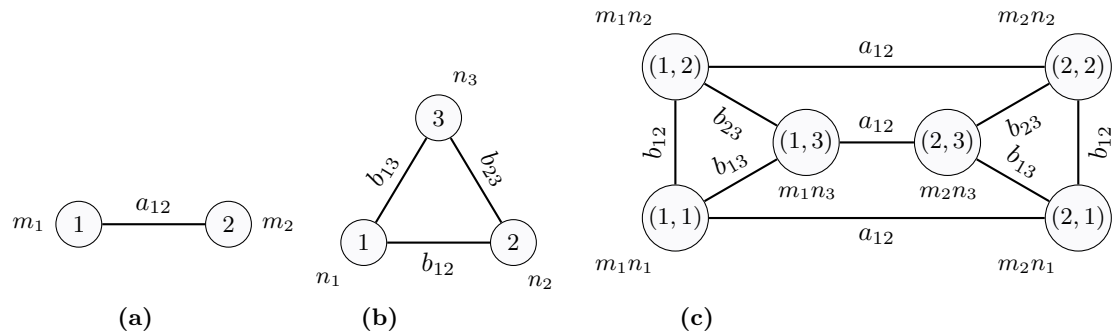


Figure 3.1 Node- and edge- weighted graph product: (a) A 2-node graph (b) A 3-node graph, (c) Result of the node- and edge- weighted product of (a) and (b).

$\mathbb{R}^{n_2\times n_2}$, *respectively. The node- and edge-weighted graph* $\mathcal{G} = \mathcal{G}_1\square\mathcal{G}_2$ *has the vertex set* $\mathcal{V} = \mathcal{V}_1\times\mathcal{V}_2$ *with the diagonal node-weight matrix* $M = M_1 \otimes M_2$. *Two vertices* (i_1, i_2), $(j_1, j_2) \in \mathcal{V}$ *are adjacent if and only if either* $i_1 = j_1$ *and* $i_2j_2 \in \mathcal{E}_2$ *or* $i_2 = j_2$ *and* $i_1j_1 \in \mathcal{E}_1$.

Remark 3.3.1. *The asymmetric Laplacian of the node- and edge-weighted graph* $\mathcal{G} = \mathcal{G}_1\square\mathcal{G}_2$ *is*

$$L_m = M^{-1}L_{\mathcal{G}} = \left(M_1^{-1}L_{\mathcal{G}_1}\right) \otimes M_2^{-1} + M_1^{-1} \otimes \left(M_2^{-1}L_{\mathcal{G}_2}\right). \tag{3.13}$$

An example of the node- and edge-weighted product is shown in Fig. 3.1. Unfortunately, the eigenvalue and eigenvector relationships in (3.10) and (3.11) no longer apply for L_m in (3.13). However, using the symmetrized Laplacian from (3.6) and a first-order approximation of the resulting eigenvalues and corresponding eigenvectors we can generalize Lemma 3.3.1 as follows.

Lemma 3.3.2. *Let* $\mathcal{G}_1$ *and* $\mathcal{G}_2$ *be node- and edge-weighted graphs with diagonal node-weight matrices* $M_1 \in \mathbb{R}^{n_1\times n_1}$ *and* $M_2 \in \mathbb{R}^{n_2\times n_2}$. *Define* $L_1 = L_{\mathcal{G}_1}$, $L_{m_1} = M_1^{-1}L_1$, $L_2 = L_{\mathcal{G}_2}$, *and* $L_{m_2} = M_2^{-1}L_2$. *Let* $\lambda_1 \leq \cdots \leq \lambda_{n_1}$ *and* $\bar{\lambda}_1 \leq \cdots \leq \bar{\lambda}_{n_1}$, *denote the eigenvalues of* L_1 *and* L_{m_1}, *respectively. Let* $\mu_1 \leq \cdots \leq \mu_{n_2}$ *and* $\bar{\mu}_1 \leq \cdots \leq \bar{\mu}_{n_2}$, *denote the eigenvalues of* L_2 *and* L_{m_2}, *respectively. If* $\mathcal{G}_1$ *is* p*-area complete and the following are satisfied:* $\lambda_p < \mu_2$, *and* $a\bar{\lambda}_p < \bar{\mu}_2$, *where* $a = n_2/tr(M_2)$, *then the node- and edge-weighted graph* $\mathcal{G} = \mathcal{G}_1\square\mathcal{G}_2$ *is* p*-area complete.*

Proof. See Appendix A.2. □

Remark 3.3.2. *An arbitrarily large* p*-area complete graph may also constructed using Lemma 3.3.1 and then choosing the node weights in a way that preserves the* p*-area complete property.*

Lemmas 3.3.1 and 3.3.2 are nicely interpreted in terms of the time-scale separation of the system (3.1a) detailed in Section 2.4. Here, the NDS with network graph $\mathcal{G}$ can be partitioned according to the p areas of $\mathcal{G}_1$, but where each node in $\mathcal{G}_1$ has been replaced by a copy of

$\mathcal{G}_2$. The assumptions that $\lambda_p < \mu_2$ and $a\bar{\lambda}_p < \bar{\mu}_2$, respectively, guarantee that the dynamics associated with the nodes in $\mathcal{G}_2$ reach a consensus faster than the slow modes associated with $\lambda_k,\ k = 1, \ldots, p$, after which each copy of $\mathcal{G}_2$ can be considered as a single aggregate node. This will be further illustrated in the simulation example in Section 4.4.

Chapter 4

Input Localization in Second-Order Synchronization Networks

We next extend the results from Chapter 3 to second-order consensus networks, also called *synchronization networks*. The practical motivation for studying such models comes from the swing model of power system networks (cf. Appendix C). Consider a second-order consensus network defined over the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where the dynamic model of the overall network can be written as

$$M\ddot{x}(t) = -Lx(t) - D\dot{x}(t) + \tilde{B}\tilde{u}(t), \tag{4.1}$$

where $x(t) \in \mathbb{R}^n$, $\tilde{u}(t) \in \mathbb{R}^{\tilde{m}}$, M, $L = L_{\mathcal{G}}$, and $\tilde{B}$ are defined as in (3.1a), and $D = \text{diag}[d_1, \ldots, d_n]$, with $d_i \geq 0,\ \forall i \in \mathcal{V}$, is a real-valued damping coefficient. Our objective is to localize the clusters in which elements of $\tilde{u}(t)$ enter using chosen measurements of $x(t)$ and $\dot{x}(t)$. In this section, we show that Algorithm 1 can be naturally extended for this purpose if the system is *weakly damped*, defined as:

Definition 4.0.3. *The system in (4.1) is said to be weakly damped when [TM01]*

$$(D\psi, \psi) - 4(M\psi, \psi)(L\psi, \psi) < 0, \tag{4.2}$$

where $D = diag[d_1, \ldots, d_n]$ and $\psi \in \mathbb{C}^n$ is any non-zero vector.

Denoting $\xi_1 = x$ and $\xi_2 = \dot{x}$, we can write (4.1) as

$$\begin{bmatrix} \dot{\xi}_1(t) \\ \dot{\xi}_2(t) \end{bmatrix} = \underbrace{\left[\begin{array}{c|c} 0 & I_n \\ \hline -L_m & -M^{-1}D \end{array}\right]}_{A} \begin{bmatrix} \xi_1(t) \\ \xi_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1}\tilde{B} \end{bmatrix} \tilde{u}(t), \tag{4.3}$$

Since A in (4.3) has dimension $2n$ with complex eigenvalues and eigenvectors, we cannot directly apply Definition 2.2.1 to the eigenvectors of A to derive a relationship between the nodal domains of $\mathcal{G}$ and the system-theoretic properties of (4.3). However, with a slight abuse of notation we will define the nodal domains of A in (4.3) to be the nodal domains of $\mathrm{Re}(\nu_k)$, where $\nu_k \in \mathbb{C}^n$ is an (right) eigenvector of (4.1) corresponding to eigenvalue $\mu_k \in \mathbb{C},\ k = 1, \ldots, n$. In the next section, we present a lemma followed by a theorem that motivate this definition by relating ν_k to the eigenvectors of A in (4.3).

4.1 Nodal Domains of Weakly Damped Second-order Networks

We start by defining a quadratic eigenvalue problem (QEP) for (4.1) as finding scalars (eigenvalues) $\mu \in \mathbb{C}$ and vectors (right eigenvectors) $\nu \in \mathbb{C}^n$ that satisfy

$$(\mu^2 M + \mu D + L)\nu = 0, \tag{4.4}$$

where $M,\ L$, and D are in general $n \times n$ complex matrices. Solutions to (4.4) are largely dependent on the structure and properties of $M,\ L$, and D [TM01]. Since $M > 0,\ D > 0$, and L is a Laplacian matrix, we can state the following results.

Lemma 4.1.1. *Let $\mu_k \in \mathbb{C}$ and $\nu_k \in \mathbb{C}^n$ be a solution to the QEP for (4.1). Let $\psi_k \in \mathbb{C}^{2n}$ be an eigenvector of A in (4.3) corresponding to eigenvalue μ_k. Then $\psi_k^T = \begin{bmatrix}\nu_k^T & \mu_k \nu_k^T\end{bmatrix}$.*

Proof. Let $\psi_k = \begin{bmatrix}\psi_{k,1}^T & \psi_{k,2}^T\end{bmatrix}^T$, where $\psi_{k,1},\ \psi_{k,2} \in \mathbb{C}^n$. Solving $(A - \mu_k I_{2n})\psi_k = 0$ yields

$$\psi_{k,2} - \mu_k \psi_{k,1} = 0 \tag{4.5}$$

$$-L_m \psi_{k,1} - M^{-1} D \psi_{k,2} - \mu_k \psi_{k,2} = 0 \tag{4.6}$$

Substituting (4.5) into (4.6) and pre-multiplying by M yields

$$L\psi_{k,1} + \mu_k D \psi_{k,1} + \mu_k^2 M \psi_{k,1} = 0, \tag{4.7}$$

which is a solution to the QEP for (4.1). Hence $\psi_{k,1} = \nu_k$. □

The next theorem uses Lemma 4.1.1 to relate the nodal domains of A in (4.3) to those of L_m.

Theorem 4.1.1. *Let $\psi_k \in \mathbb{C}^{2n}$ be a right eigenvector of A in (4.3) corresponding to eigenvalue μ_k. Let $v_k \in \mathbb{R}^n$ be a right eigenvector of L_m corresponding to λ_k. If D is small enough, then the nodal domains of A in (4.3) are identical to those of L_m.*

Proof. As $d_i \to 0,\ \forall i \in \mathcal{V}$, then $\mu_k \to \pm\sqrt{-\lambda_k}$. From (4.7) in Lemma 4.1.1 we have

$$L\nu_k = -\mu_k^2 M\nu_k - \mu_k D\nu_k \to \lambda_k M\nu_k. \tag{4.8}$$

When $D \to 0$, (4.8) becomes the generalized eigenvalue problem (GEP) for the pair (L, M), which implies $\nu_k \to v_k$, $k = 1, \ldots, n$. □

Remark 4.1.1. *The left eigenvectors $\phi_k \in \mathbb{C}^{2n}$ of (4.3) are related to the left eigenvectors w_k of L_m through an analogous construction.*

4.2 Residues of Second-Order Systems

As in Section 3.1, for analysis purposes we assume every component of $\tilde{u}(t)$ in (4.3) to be impulsive, and consider the outputs as

$$y(t) = \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \begin{bmatrix} \xi_1(t) \\ \xi_2(t) \end{bmatrix}, \tag{4.9}$$

where $C_1 \in \mathbb{R}^{h\times n}$ and $C_2 \in \mathbb{R}^{h\times n}$ have rows of the form $\mathbf{e}_j^T,\ j \in \mathcal{S}$, with $\mathcal{S} \subseteq \mathcal{V},\ |\mathcal{S}| = h$. Since we now have two outputs from every node $j \in \mathcal{S}$, we can write the residues of (4.3) as $R_{ij}^k = \begin{bmatrix} Q_{ij}^k & P_{ij}^k \end{bmatrix}^T$, where Q_{ij}^k, $P_{ij}^k \in \mathbb{C}$ are defined as

$$Q_{ij}^k = [\mathbf{e}_j^T\ 0]\psi_k\phi_k^* M^{-1} \begin{bmatrix} 0 \\ \mathbf{e}_i \end{bmatrix}, \qquad P_{ij}^k = [0\ \mathbf{e}_j^T]\psi_k\phi_k^* M^{-1} \begin{bmatrix} 0 \\ \mathbf{e}_i \end{bmatrix}, \tag{4.10}$$

where * denotes the conjugate transpose. Lemma 4.1.1 and Theorem 4.1.1 together imply the following corollary relating the nodal domains of A in (4.3) to the residues defined in (4.10).

Corollary 4.2.1. *Q_{ij}^k and P_{ij}^k from (4.10) satisfy*

$$-sgn(Im(Q_{ij}^k)/Im(\mu_k)) = sgn([v_k]_j[w_k]_i), \tag{4.11}$$

$$sgn(Re(P_{ij}^k)) = sgn([v_k]_j[w_k]_i). \tag{4.12}$$

Corollary 4.2.1 provides the theoretical extension of Lemma 3.1.3 to second-order systems of the form (4.1), where each node has two dynamic states. Corollary 4.2.1 implies that the central network manager is oblivious to the distinction between first- and second-order networks, and he may construct the table of localization keys according to the nominal Laplacian matrix $\bar{L}_m$ as before. However, the local operators now have a choice of measurements with which they may run Algorithm 1, namely $\xi_1(t)$ and $\xi_2(t)$; the only modification of the algorithm is that $\hat{\mathcal{K}}_{ij}$

should be constructed according to (4.11) or (4.12), respectively.

4.3 Localizing the Resonant Input

The remainder of the chapter is devoted to the specific application of localizing forced harmonic inputs of the form $u(t) = f_0 \sin \omega_f t$. Our basic approach is as follows. Recalling Assumption 3.0.1, the system (4.1) consists of p coherent areas, hence the inter-area dynamics are governed by $p-1$ *inter-area modes*. Using the phase angles and frequency measurements available from sensors such as Phasor Measurement Units (PMU) at a selected number of generators, we first run a system identification routine to estimate the transfer function of the entire power system model (4.1), isolate the residues of the $p-1$ slow poles, and finally construct a localization key and compare it with the table of keys $\mathcal{K}$ described previously in Chapter 3.

4.3.1 Transfer Function Derivations

We begin by analyzing the input-output transfer function to see how the effective transfer-function residues are related to the impulse-response residues used in the localization algorithm. For the sake of clarity in the following derivations, we consider a single input and a single output in the undamped system

$$\begin{bmatrix} \dot{\xi}_1(t) \\ \dot{\xi}_2(t) \end{bmatrix} = \underbrace{\left[\begin{array}{c|c} 0 & I_n \\ \hline -L_m & 0 \end{array}\right]}_{A} \begin{bmatrix} \xi_1(t) \\ \xi_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1}B \end{bmatrix} u(t), \tag{4.13}$$

where the measured state is x_1 (phase-angle). Let the input enter node k and let the output be measured at node l. We first assume that the incoming disturbance is impulsive, and run a subspace identification algorithm (see Appendix B). Since the network has p areas the identified transfer function can be written in a truncated pole-residue form as

$$\frac{Y_l(s)}{U_k(s)} = \underbrace{\sum_{i=2}^{p} \frac{R^i}{(s^2+\lambda_i)}}_{g^{slow}(s)} + \tilde{g}(s), \tag{4.14}$$

where $\tilde{g}(s)$ contains the zero modes and the fast modes, $R^i \in \mathbb{R}$ is the residue corresponding to $\lambda_i = -\mu_i^2$, and $Y_l(s)$ and $U_k(s) = 1$ are the Laplace transforms of the output and input, respectively. The actual input to the system, however, is $u(t) = f_0 \sin \omega_f t$, the Laplace transform of which is $U(s) = f_0 \omega_f / (s^2 + \omega_f^2)$. Then the actual expression for $Y_l(s)$, assuming $f_0 = 1$ and

neglecting the contribution from $\tilde{g}(s)$ without loss of generality, is

$$Y_l(s) = \left(\frac{\omega_f}{s^2+\omega_f^2}\right)\sum_{i=2}^{p}\frac{R^i}{s^2+\lambda_i} \tag{4.15}$$

$$= \frac{1}{4}\left(\frac{j\omega_f}{s+j\omega_f}+\frac{-j\omega_f}{s-j\omega_f}\right)\times \sum_{i=2}^{p}\left(\frac{P^i+jQ^i}{s+\mu_i}+\frac{P^i-jQ^i}{s-\mu_i}\right), \tag{4.16}$$

where $\pm\mu_i = \pm j\sqrt{\lambda_i}$ and $P^i \in \mathbb{R}$ and $Q^i \in \mathbb{R}$ are the real and complex parts of the residues corresponding to modes $\pm\mu_i$. From Corollary 4.2.1, it is easy to verify $P^i = 0$ and $jQ^i = \mu_i R^i$. Then (5.2) can be written as

$$Y_l(s) = \frac{1}{4}\sum_{i=1}^{p}\left[\left(\frac{j\omega_f}{s+j\omega_f}+\frac{-j\omega_f}{s-j\omega_f}\right)\left(\frac{jQ^i}{s+\mu_i}+\frac{-jQ^i}{s-\mu_i}\right)\right] \tag{4.17}$$

$$= \frac{1}{4}\sum_{i=1}^{p}\left[\frac{-\omega_f Q^i}{(s+j\omega_f)(s+\mu_i)}+\frac{\omega_f Q^i}{(s-j\omega_f)(s+\mu_i)}+ +\frac{\omega_f Q^i}{(s+j\omega_f)(s-\mu_i)}+\frac{-\omega_f Q^i}{(s-j\omega_f)(s-\mu_i)}\right]. \tag{4.18}$$

It can be shown that the output decomposes into a component corresponding to the input frequency ω_f and components corresponding to the system's slow poles as

$$Y_l(s) = \frac{1}{2}\sum_{i=1}^{p}\left[\left(\frac{-\omega_f\mu_i Q^i}{\omega_f^2-\lambda_i}\frac{1}{s+j\omega_f}+\frac{\omega_f\mu_i Q^i}{\omega_f^2-\lambda_i}\frac{1}{s-j\omega_f}\right) +\left(\frac{j\omega_f^2 Q^i}{\omega_f^2-\lambda_i}\frac{1}{s+\mu_i}+\frac{-j\omega_f^2 Q^i}{\omega_f^2-\lambda_i}\frac{1}{s-\mu_i}\right)\right]. \tag{4.19}$$

From (4.19), we see that the effective residues associated with the systems slow poles are

$$\tilde{R}^i = \frac{j\omega_f^2 Q^i}{(\omega_f^2-\lambda_i)} = \frac{\omega_f^2}{\omega_f^2-\lambda_i}\mu_i R^i, \tag{4.20}$$

for $i = 2, \ldots, p$. From (4.20) we see that not only do these effective residues grow in magnitude compared to their nominal values as the input frequency approaches the resonance condition $\omega_f^2 \to \lambda_i$, but they also change signs when $\omega_f^2 < \lambda_i$. This will be accounted for in the following input localization algorithm.

4.3.2 Input Localization Algorithm

This section details an algorithm to localize the source of a forced harmonic input in a large-scale power system. We will also discuss the ability to quickly *detect* the input from a measurement-based perspective; however, the focus of the algorithm is localization.

As detailed in Section 3.2, the algorithm for the first-order network requires minimal *a-priori* information in the form of localization keys, and the requirement is no different here. To recollect, a central operator has a nominal knowledge of the clustering structure of the network graph $\mathcal{G}$. Denote the nominal Laplacian matrix by $\overline{L}_m$. Using $\overline{L}_m$, construct a $p \times p$ array of input localization keys $\mathcal{K} = [\mathcal{K}_{ij}]$. Construction of this key is a one-time task as long as the clustering remains the same as $\overline{L}_m$. In practice, this means $\mathcal{K}$ can be used despite changes in real-time operating conditions, such as loss of generators and tie-lines, so long as the clustering structure does not change. Once this table of keys has been computed, the lth row of the table is associated with Area l (i.e. given to the local operator of Area l), which encompasses the node set $\mathcal{V}^l$. The local operator taking measurements at $q_l \in \mathcal{V}^l$ runs a system identification procedure to estimate the transfer function $Y_l(s) = \hat{g}^{slow}(s)$. If the number of slow modes detected exceeds the expected number, i.e., the transfer function contains $\hat{p} > p$ modes in the slow spectrum, then we say that the input pole has been *detected*. Once the pole has been detected, the operator attempts to *distinguish* the input from the system's natural response, after which the residue signs can be corrected according to equation (4.20). The input localization steps are summarized in Algorithm 2. It should be noted that the complexity of the first step in the algorithm—the system identification subroutine—is by far the most complex and time consuming step.

4.3.3 Distinguishing the Exogenous Mode

The unwanted input is easily *detected* in Algorithm 2 if the system identification subroutine successfully identifies ω_f in addition to the slow poles of (4.3). However, distinguishing the unwanted mode ω_f from $\{\lambda_i, \ldots, \lambda_p\}$ requires additional information and additional computation. In this section we consider the two cases where $\omega_f^2 \neq \lambda_i, \forall i = 1, \ldots p$, and where $\omega_f^2 \to \lambda_i$ for some $i,\ i = 2, \ldots, p$.

First we consider these cases in the weakly damped system (4.3). The presence of any damping in the natural frequencies is enough information to distinguish ω_f which is assumed to have zero damping. The distinguishing step (line 7 in Algorithm 2) can simply include an all-to-all comparison of $Re(\mu_i)$. In the first case, when $\omega^2 \neq \lambda_i, \forall i = 1, \ldots p$, this procedure results in an additional $\binom{\hat{p}}{2}$ operations. In the second case, as ω_f approaches the resonance condition, this procedure can be reduced to a single additional operation as the candidate modes have been previously identified in the algorithm (line 2).

In a second scenario, consider the response of the undamped system (4.13). Here damping

Algorithm 2 Localization Algorithm

Require: Measurements $y_{q_l}(t_k),\ t_k = t_0,\ t_1, \ldots, t_\varsigma$ and $\mathcal{K}_l = \{\mathcal{K}_{l1}, \ldots, \mathcal{K}_{lp}\}$

```
Run System Identification to estimate (Â, B̂, Ĉ)
Detect slow modes λ̂_i = −(μ̂_i)^2, i = 2, ..., p̂ such that λ̂_p̂ ≪ λ̂_{p̂+1}
Estimate effective residues R̃^i, i = 1, ..., p̂
if p̂ > |K_l| + 1 = p then
    for l = 1 → p̂ − p do
        Distinguish ω_f from natural fequencies
        if Distinguishable then
            Estimate nominal residues R̂^i from (4.20)
        end if
    end for
end if
Construct localization key
    K̂_{lκ} = {sgn(R̂^2), ..., sgn(R̂^p)}
for i = 1 → p do
    Match the estimated key to stored table
    if Match, i.e., K_{lκ} = K_{li} then
        Input may be located in Area i
    end if
end for
```

cannot be used as a distinguishing factor. However, if one has *a-priori* information about the *magnitude* of the nominal residues R^i and $\omega_f^2 \neq \lambda_i, \forall i = 2, \ldots, p$, then one could formulate a non-linear least-squares problem using this information with the effective residues described by (4.20) to determine ω_f from a set of candidates. Namely,

$$\omega_f = \arg\min_{\omega} \|\tilde{R}^k - f_k(\omega; \lambda_2, \ldots, \lambda_p)\| \tag{4.21}$$

where

$$f_k(\omega; \lambda_2, \ldots, \lambda_p) = \frac{\omega^2}{(\omega - \lambda_i)} \mu_i R^k, \tag{4.22}$$

and R^k is assumed known. However, as $\omega_f^2 \to \lambda_i$, the effective residue of the input also approaches the magnitude of its λ_i counterpart, making it impossible to distinguish ω_f from λ_i . The 'distinguish' step in Algorithm 2, therefore, works best when the system is weakly damped.

4.4 Numerical Examples

As mentioned in the introduction, our interest in the model (4.3) is motivated by input localization problems in large-scale power systems, where disturbances must be localized in real-time using measurements of voltage, phase angle, and frequency [AF03]. The recent outburst of the Wide-Area Measuring Systems (WAMS) technology [Pha83] has inspired an urgent need for such disturbance localization using Phasor Measurement Units (PMUs). The approach proposed in this chapter is highly suitable for WAMS. Unlike the model-based observer designs presented in [Sha12b; Pas13], we do not necessarily require access to all of the measurements communicated across the network. We next implement Algorithm 1 on a 60-generator, 292-tie-line, 4-area power system model using outputs from only a single measurement node. We also demonstrate Algorithm 1 with a well-defined 5-area model of the U.S. west coast power grid, also referred to as the Western Electricity Coordinating Council (WECC).

4.4.1 Dynamic Model of the 60-machine Power System

For completeness, we recapitulate the second-order swing model of a power system network. More details on graph theoretic modeling of power system networks are provided in Appendix C Consider the 60-generator power system model defined over the graph $\mathcal{G}$ shown in Fig. 4.1(a). Each node represents a synchronous generator and each edge represents a transmission line. For simplicity, we assume that the lines are directly connected between the generator buses. Ignoring line-loss, the dynamic electro-mechanical model of the ith generator can be written as [AF03]

$$\dot{\delta}_i = \omega_i - \omega_s, \tag{4.23}$$

$$m_i\dot{\omega}_i = P_i - d_i\omega_i - \sum_{k\in\mathcal{N}_i} \frac{E_iE_k\sin(\delta_i - \delta_k)}{x_{ik}}, \tag{4.24}$$

where δ_i is the internal rotor angle, $\omega_s = 120\pi$ is the synchronous speed for a 60 Hz system, ω_i is the rotor angular velocity, $m_i = 2H_i$ denotes the electro-mechanical inertia, P_i is the mechanical power, d_i is the internal machine damping, E_i is the magnitude of the voltage phasor, x_{ik} denotes the reactance of the tie-line connecting the ith and k^{th} machines, and $\mathcal{N}_i$ is the neighbor set of the ith generator. We linearize (4.23)–(4.24) about an initial equilibrium (δ_{i0}, ω_s) where $0 < \delta_{i0} < 90^\circ \ \forall\, i \in \mathcal{V}$. Denoting the perturbed state variables as $\Delta\delta = [\Delta\delta_1 \ \ldots \ \Delta\delta_{60}]^T$ and $\Delta\omega = [\Delta\omega_1 \ \ldots \ \Delta\omega_{60}]^T$, and assuming P_i to be constant for all i, the LTI model with a disturbance input $\tilde{u}(t)$ entering the jth generator can be written as

$$\left[\begin{array}{c} \Delta\dot{\delta} \\ \hline \Delta\dot{\omega} \end{array}\right] = \left[\begin{array}{c|c} 0 & I_n \\ \hline -L_m & M^{-1}D \end{array}\right] \left[\begin{array}{c} \Delta\delta \\ \hline \Delta\omega \end{array}\right] + \left[\begin{array}{c} 0 \\ \hline M^{-1}\mathbf{e}_j \end{array}\right] \tilde{u}(t), \tag{4.25}$$

where $M = \text{diag}[m_1, \ldots, m_{60}]$, $D = \text{diag}[d_1, \ldots, d_{60}]$, $L_m = M^{-1} L_{\mathcal{G}}$, and the edge weights of $\mathcal{G}$ are $a_{ik} = \frac{E_i E_k}{x_{ik}} \cos(\delta_{i0} - \delta_{k0})$. We assume the network admits a 4-area partition, as shown in Fig. 4.1(a).

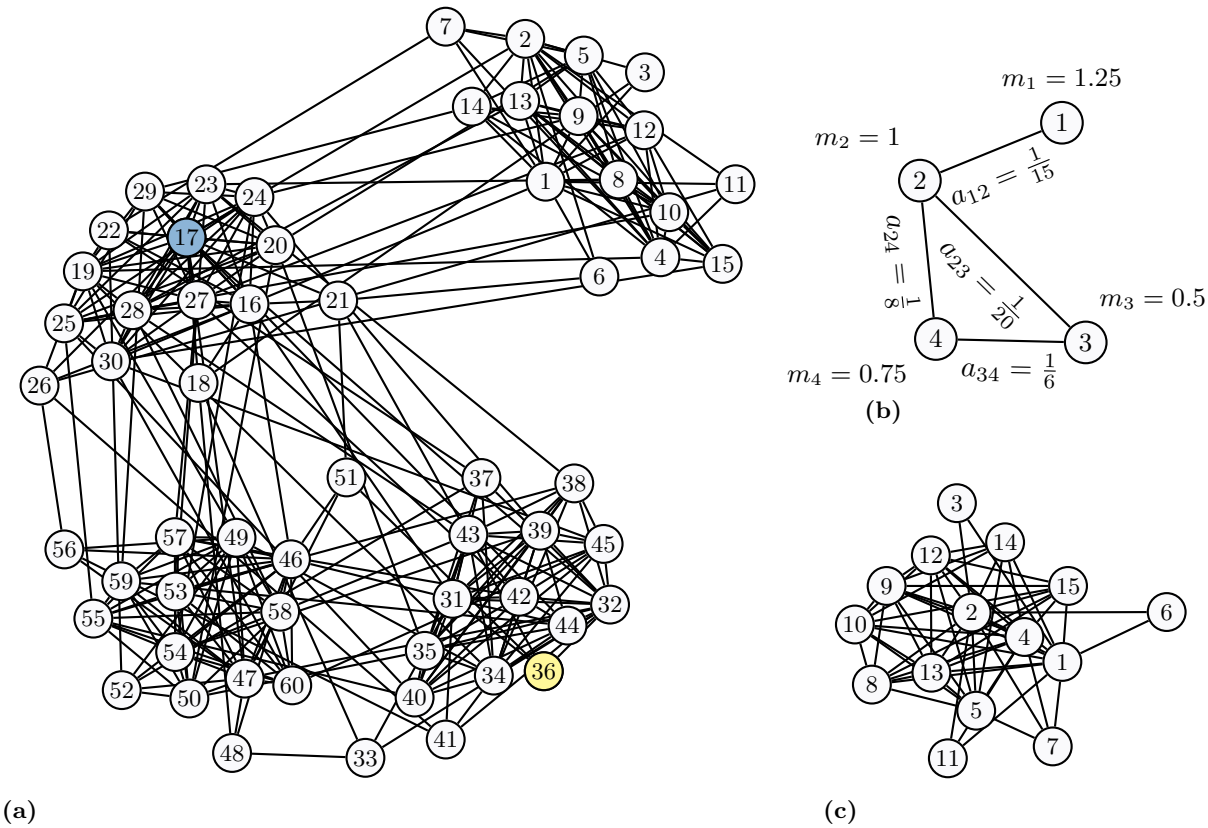


Figure 4.1 (a) Network $\mathcal{G}$ used in the numerical example. Area assignments are sequential: $\mathcal{V}^1 = \{1, \ldots, 15\}$, $\mathcal{V}^2 = \{16, \ldots, 30\}$, $\mathcal{V}^3 = \{31, \ldots, 45\}$, and $\mathcal{V}^4 = \{46, \ldots, 60\}$. Spacing between vertices is approximately (*resistance distance*)$^{1/5}$. Measurement node(s) are highlighted in blue, while the disturbance location is yellow. (b) The sparse, 4-node graph $\mathcal{G}_1$, with node and edge weights labeled. (c) A dense, random, 15-node graph $\mathcal{G}_2$. Edge weights are 10 per-unit and node weights are unity.

4.4.2 Construction of Nominal Keys

The underlying network graph $\mathcal{G}$ (Fig. 4.1(a)) is constructed by taking the node- and edge-weighted Cartesian product (Definition 3.3.2) of the two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ shown in Fig. 4.1(b) and Fig. 4.1(c), respectively. The 4-node, node-and edge-weighted graph $\mathcal{G}_1$ is *sparse*, and hence the aggregate structure of $\mathcal{G}$ closely resembles the structure of $\mathcal{G}_1$. The topology of the 15-node

Table 4.1 Localization Keys for Fig. 4.1(a)

Area	1	2	3	4
1	$\cdot$	$\{-,-,-\}$	$\{-,+,-\}$	$\{-,-,+\}$
2	$\cdot$	$\cdot$	$\{+,-,+\}$	$\{+,+,-\}$
3	$\cdot$	$\cdot$	$\cdot$	$\{+,-,-\}$
4	$\cdot$	$\cdot$	$\cdot$	$\cdot$

Table 4.2 Localization keys for the WECC model Fig. 4.3(a)

Area	1	2	3	4	5
1	$\cdot$	$\{+,-,-,-\}$	$\{-,-,+,+\}$	$\{-,+,+,-\}$	$\{-,+,-,+\}$
2	$\cdot$	$\cdot$	$\{-,+,-,-\}$	$\{-,-,-,+\}$	$\{-,-,+,-\}$
3	$\cdot$	$\cdot$	$\cdot$	$\{+,-,+,-\}$	$\{+,-,-,+\}$
4	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\{+,+,-,-\}$
5	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$

random graph $\mathcal{G}_2$ is dense and arbitrary. Since the graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ satisfy the hypothesis of Lemma 3.3.2 with $p = 4$, we may construct the nominal table of localization keys using the nodal domains of $\mathcal{G}_1$. This is easily accomplished following the procedure outlined in the example in Section 3.1.3. The table of nominal keys is shown in Table 4.1. After the network topology has been constructed and the array of nominal localization keys has been computed, we simulate a disturbance in the network.

4.4.3 Impulsive Disturbance

At $t = 0$ a disturbance occurs at node 36 (in Area 3). Following the disturbance, the operator in Area 2 uses measurements of $\Delta\delta_{17}(t)$ corrupted by white Gaussian noise with a 40 dB signal-to-noise ratio recorded over $t = 0, \ldots, 60$ seconds to identify the transfer function

$$\hat{g}^{slow}(s) = \frac{0.0518s^5 - 0.0659s^4 + 0.0375s^3 - 0.0501s^2 - 0.0016s + 0.003}{s^6 + 0.0398s^5 + 1.1205s^4 + 0.0307s^3 + 0.3223s^2 + 0.0052s + 0.01778} \tag{4.26}$$

$$= \frac{-0.0085 \mp 0.0673j}{s - (-0.0091 \pm 0.2683j)} + \frac{0.0427 \pm 0.1106j}{s - (-0.0036 \pm 0.5972j)} + \frac{-0.0083 \mp 0.0169j}{s - (-0.0072 \pm 0.8314j)}. \tag{4.27}$$

The plot of $\Delta\delta_{17}(t)$ together with the impulse response of the identified transfer function is shown in Fig. 4.2(a). It can be seen that the matching is robust to the measurement noise. This operator also has access to the nominal localization keys $\mathcal{K}_2 = [\{-,-,-\}, \{+,+,+\}, \{+,-,+\}, \{+,+,-\}]$ from Table 4.1. Using Corollary 4.2.1 he constructs an estimated localization key by comparing the sign of the imaginary part of each residue in (4.27) to that of the imaginary part of the corresponding pole. The estimated key is obtained as $\hat{\mathcal{K}}_{\kappa 2} = \{+,-,+\}$. This key matches $[\mathcal{K}_2]_3$,

indicating that disturbance entered in Area 3, which is indeed correct.

Similarly, if the operator uses measurements of $\Delta\omega_{17}(t)$, the identified transfer function is

$$\hat{g}^{slow}(s) = \frac{-0.0746s^5 - 0.0182s^4 - 0.0569s^3 - 0.0172s^2 + 0.00196s - 0.00093}{s^6 + 0.0202s^5 + 1.1199s^4 + 0.0161s^3 + 0.3215s^2 + 0.0026s + 0.01761} \tag{4.28}$$

$$= \frac{0.0163 \mp 0.0016j}{s - (-0.0044 \pm 0.2672j)} + \frac{-0.0670 \pm 0.0245j}{s - (-0.0030 \pm 0.5970j)} + \frac{0.0134 \mp 0.0069j}{s - (-0.0027 \pm 0.8318j)}, \tag{4.29}$$

for which the estimated key is obtained, again, as $\hat{\mathcal{K}}_{i2} = \{+, -, +\}$, clearly showing that the choice of measurement does not affect the result of localization.

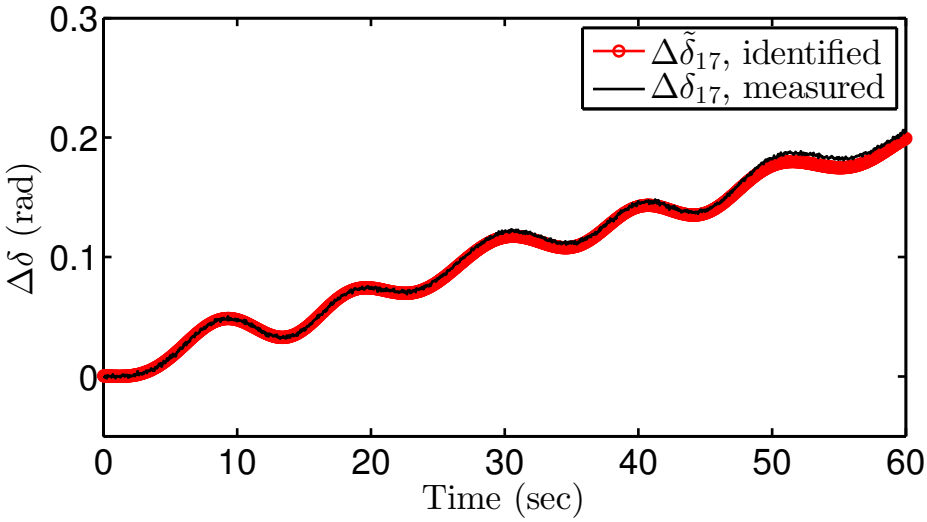
4.4.4 U.S. West Coast Power Grid Simulation

We next demonstrate our algorithm on a 5-area WECC shown in Figure 4.3(a). This is a widely used model for oscillation analysis in the WECC, where the five areas represent four balancing authorities [Bal04]. This model was recently developed in collaboration with Southern California Edison using real PMU measurements over a range of disturbance events [Cha13]. The details of the model and its parameter values can be found in [Cha13]. A 3-phase line-to-ground fault is injected into the system at Bus 3. The frequency at Bus 1 is measured, and its slow modal components are extracted using ERA, as shown in Figure 4.3(b). The estimated localization key resulting from Algorithm 1 is $\{-, -, +, +\}$. It can be clearly seen from Table 4.2 that this key corresponds to a disturbance at Bus 3. Moreover, from the model parameters given in [Cha13] this model can be verified to be p-area complete with $p = 5$. Hence, the localization is unique.

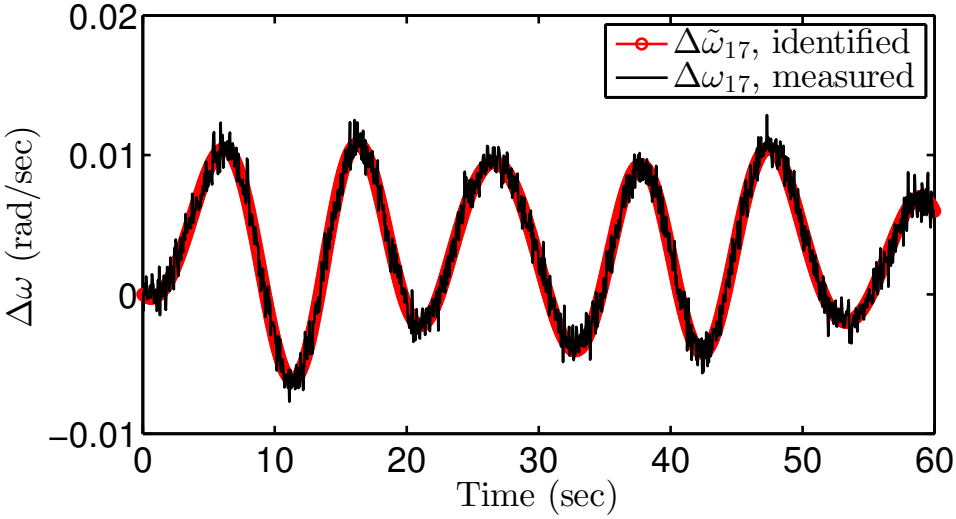
4.4.5 Localizing Forced Input

This simulation uses the same 60-generator, four-area model presented in Section 4.4.1 and shown in Fig. 4.1. The array of localization keys for this simulation is given in Table 4.1. We next simulate a disturbance input $\tilde{u}(t) = \sin \omega_f t$ with $|\omega_f^2 - \lambda_3| \leq 0.001$. The forcing function enters at node 32 in Area 3 (highlighted in yellow). Output responses from a node in each area are shown in Fig. 4.4(a) to illustrate the effects of the resonance condition as measured by local system operators. The specific nodes are chosen so that Assumption 3.2.1 has been met.

Finally, we illustrate how a particular local operator, in this case in Area 1, proceeds through Algorithm 2. A system identification is run using both state measurements from node 4, resulting in a matrix of two transfer functions. The recovered phase angle and frequency are compared against the measurements in Fig. 4.3. The close matching indicates that the system identification from the measurements is accurate (Assumption 3.2.1 has been met). Next, the system poles are distinguished from the forcing frequency ω_f. The poles and corrected residues are shown in Table 4.3 and Tables 4.4. Next, two estimated localization keys are built by applying Corollary 4.2.1. In both cases the estimated key is $\{-, +, -\}$. Comparing the estimated key to

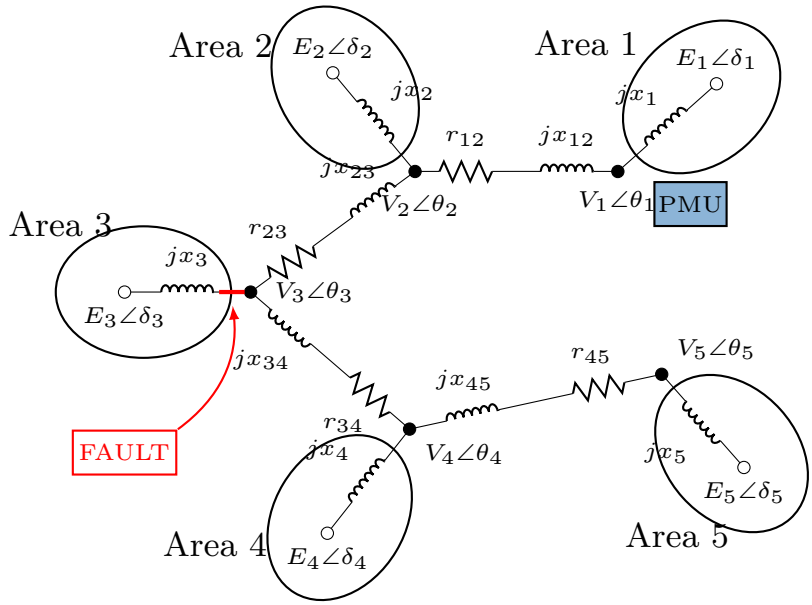


(a)

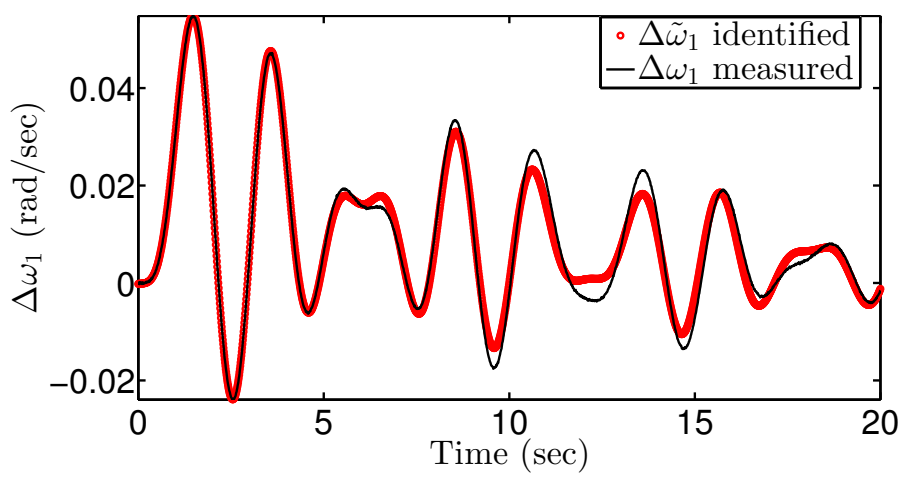


(b)

Figure 4.2 Output measurements compared to system identification from node 17: (a) Phase-angle, (b) Frequency.



(a)



(b)

Figure 4.3 WECC simulation: (a) WECC 5-area model topology, (b) Frequency measurements from Area 1.

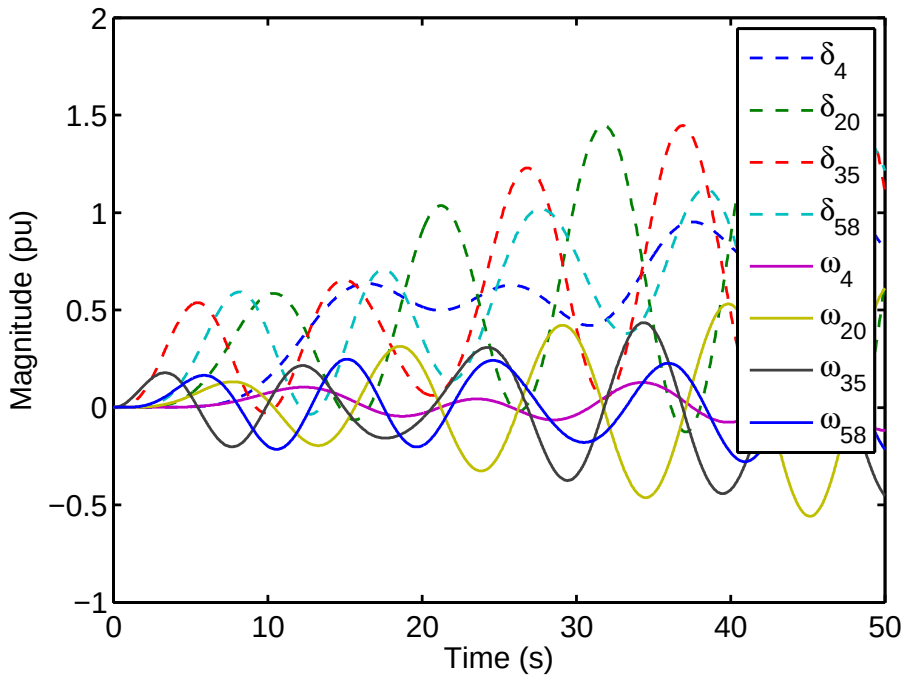
Table 4.3 Poles and Residues Measuring δ_4

μ_i	$Im(\tilde{R}^i)$
$-0.0150 + 0.2676i$	1.9801
$-0.0150 - 0.2676i$	-1.9801
$-0.0149 + 0.5964i$	-1.5351
$-0.0149 - 0.5964i$	1.5351
$-0.0230 + 0.8297i$	0.0128
$-0.0230 - 0.8297i$	-0.0128

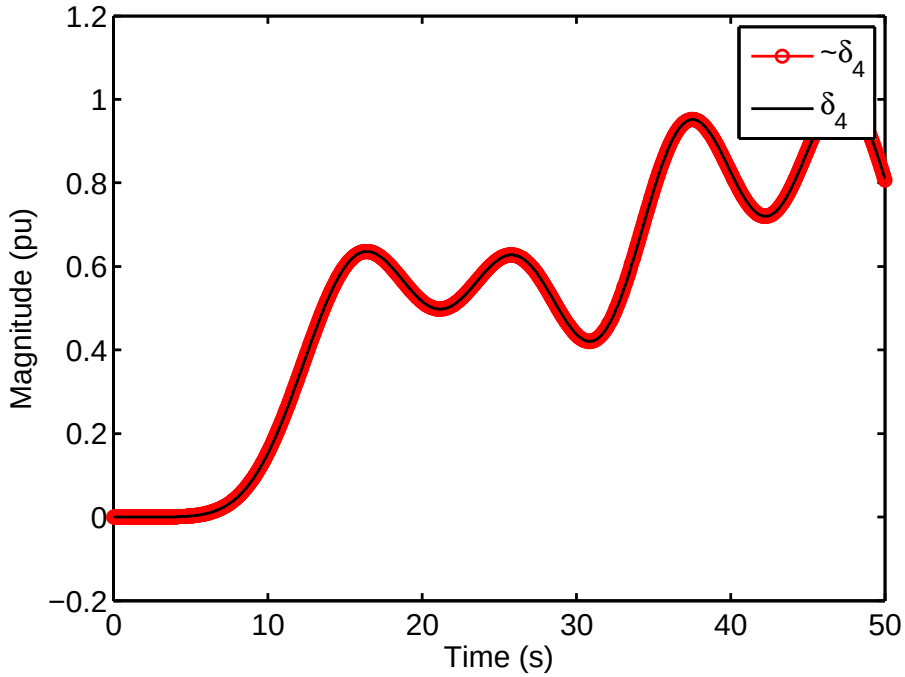
Table 4.4 Poles and Residues Measuring ω_4

μ_i	$Re(\tilde{R}^i)$
$-0.0150 + 0.2676i$	-0.5291
$-0.0150 - 0.2676i$	-0.5291
$-0.0149 + 0.5964i$	0.9518
$-0.0149 - 0.5964i$	0.9518
$-0.0230 + 0.8297i$	-0.0107
$-0.0230 - 0.8297i$	-0.0107

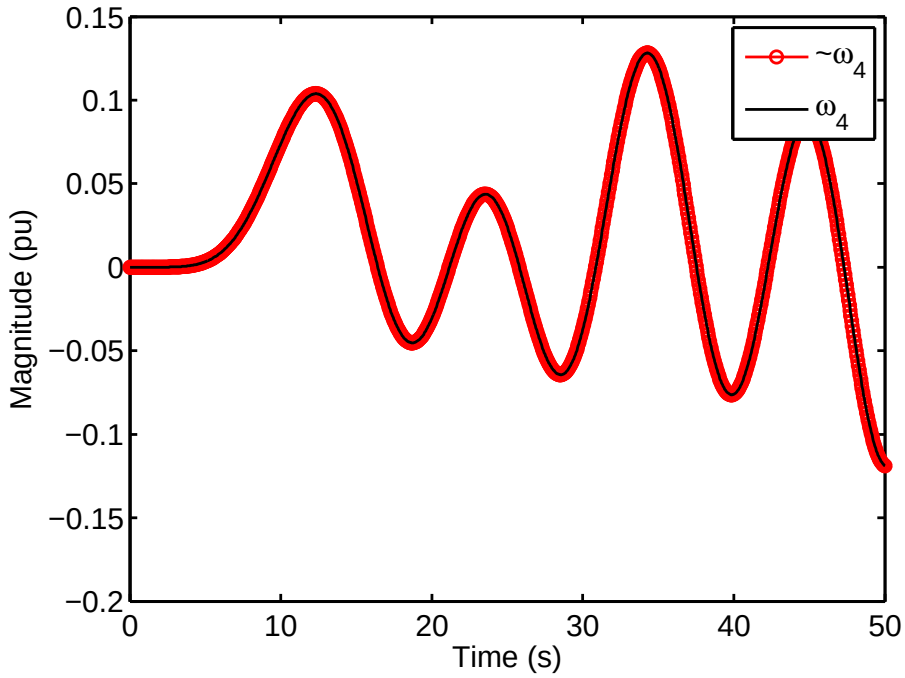
the first row of Table 4.1 (corresponding to outputs from Area 1), the operator finds a match in column 3 and only in column 3. This indicates that the input must have occurred in Area 3—which is indeed the case.


Magnitude (pu)
Time (s)
δ4
δ20
δ35
δ58
ω4
ω20
ω35
ω58

(a)


Magnitude (pu)
Time (s)
~δ4
δ4

(b)



(c)

Figure 4.3 (a) Frequency and phase angle response from points in each area. (b) and (c) Show, respectively, the phase angle and frequency response (black) of node 4 in $\mathcal{V}^1$ (Area 1) compared to the estimated states (red) obtained from a system identification algorithm.

Chapter 5

Real-time Disturbance Localization

This chapter utilizes the results from the previous two chapters to develop a real-time attack localization algorithm for power system networks, such as those modeled in Appendix C. The basic purely-differential linearized small signal model of a power system with n^g generators is

$$\left[\begin{array}{c}\Delta\dot{\delta}\\ \hline \Delta\dot{\omega}\end{array}\right] = \underbrace{\left[\begin{array}{c|c}0_{n\times n} & \omega_s I_{n\times n}\\ \hline -L_m & -M^{-1}D\end{array}\right]}_{A}\left[\begin{array}{c}\Delta\delta\\ \hline \Delta\omega\end{array}\right] + \underbrace{\left[\begin{array}{c}0\\ \hline M^{-1}e_l\end{array}\right]}_{B}u(t), \tag{5.1}$$

where $\Delta\delta = [\Delta\delta_1, \ldots, \Delta\delta_{n^g}]^T$, $\Delta\omega = [\Delta\omega_1, \ldots, \Delta\omega_{n^g}]^T$, $M = \text{diag}[m_1, \ldots, m_{n^g}]$, and $D = \text{diag}[d_1, \ldots, d_{n^g}]$. This formulation assumes the mechanical power input P_i^M to be constant for all $i \in \mathcal{V}$ so that it does not enter (5.1), however, for simplicity we consider a single external disturbance $\tilde{u}(t) \in \mathbb{R}$ entering the network through the acceleration equation of the lth generator, where the location l is unknown. Let $\mathcal{S} \subset \mathcal{V}$ denote a subset of $h \leq n^g$ generator buses. We consider the output equations as

$$y(t) = C\Delta\delta, \tag{5.2}$$

where $y(t) = [y_1(t), \ldots, y_h(t)]^T$ and rows of C are of the form $e_k^T,\ k \in \mathcal{S}$. Ideally we would like to identify the input location l directly, but as indicated in the introduction, achieving this in real-time for large power networks may be infeasible. Instead we assume the network to be divided into p coherent areas with disjoint sets of generators, denoted $\mathcal{V}^k$ with $k = 1, \ldots, p$. We take advantage of the time-scale separation properties of (C.6) to formulate our problem as follows:

Following an attack on the network, rapidly localize the area(s) in which the attack input $\tilde{u}(t)$ may have entered using real-time Synchrophasor measurements $y(t)$ from (5.2).

To solve this problem, we propose the following three steps:

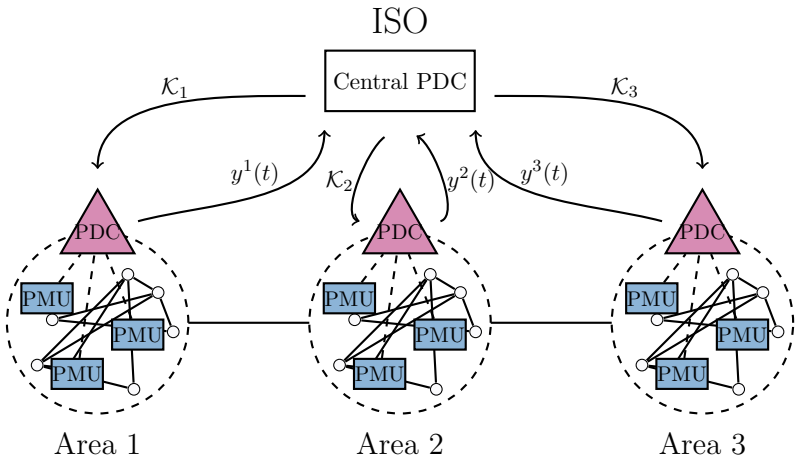


Figure 5.1 Complete cyber-physical architecture for the real-time localization algorithm. Step 1 and Step 2 are carried out by the central ISO, while Step 3 is carried out by the local areas independently.

1. Assuming that each of the p coherent areas can be represented by an equivalent aggregate generator connected over an equivalent topology $\mathcal{G}^s$ with p nodes, develop a real-time algorithm by which $\mathcal{G}^s$ can be identified using $y(t)$.

2. Construct a set of *localization keys*—to be described in Section 5.1.2—based on the identified graph $\mathcal{G}^s$ which maps input-output locations in the network to the residues of the transfer function of the equivalent model.

3. Develop a real-time localization algorithm that utilizes the keys from Step 2 by

   (a) Estimating the poles and the residues of the transfer function of (5.1)–(5.2) via a system identification algorithm using $y(t)$.

   (b) Exploit the same relationship between the poles and residues of the reduced-order equivalent system explained in Section 5.1.2 to build an estimated localization key.

   (c) Compare the estimated localization key to the set of keys provided in Step 2 to localize the area(s) in which the input $\tilde{u}(t)$ may have entered.

The overall cyber-physical architecture for implementing these steps is shown in Fig. 5.1. In this figure, the power system is assumed to be divided into three coherent clusters, denoted as Area 1, Area 2, and Area 3. Each area is equipped with a control center of its own, here represented simply by a Phasor Data Concentrator (PDC). PMUs in each area transmit measurements in real-time over a local area network to the local PDC where Step 3 is carried out. These measurements (or a select subset of them) are sent to the control center of the central ISO, where Steps 1 and 2 are carried out in parallel to Step 3a. After Step 2 has been completed,

the necessary information is transmitted back to the local areas so that they may complete Steps 3b–3c. We next describe each of these steps in detail.

5.1 Steps for Real-time Disturbance Localization

5.1.1 Step 1—Online Topology Identification

This step is carried out by the central system operator using real-time PMU measurements $y(t)$ in (5.2) streaming from each area. Here we must assume that the p-area equivalent topology $\mathcal{G}^s$ exists, and that the set of measured phase angles at the generator buses in $\mathcal{S}$ ensures that $\mathcal{G}^s$ is identifiable. Each coherent area can be represented by an equivalent generator, whose rotor angles and angular velocities are defined as $\delta^s = [\delta_1^s, \ldots, \delta_p^s]$ and $\omega^s \triangleq \dot{\delta}^s/\omega_s$. The linearized small-signal dynamics of the equivalent model can be written analogously to (5.1) as

$$\left[\begin{array}{c}\Delta\dot{\delta}^s \\ \hline \Delta\dot{\omega}^s\end{array}\right] = \underbrace{\left[\begin{array}{c|c}0_{p\times p} & \omega_s I_{p\times p} \\ \hline -L_m^s & -(M^s)^{-1}D^s\end{array}\right]}_{A^s}\left[\begin{array}{c}\Delta\delta^s \\ \hline \Delta\omega^s\end{array}\right] + \underbrace{\left[\begin{array}{c}0 \\ \hline (M^s)^{-1}\mathbf{e}_l\end{array}\right]}_{B^s}\tilde{u}(t), \tag{5.3}$$

where $M^s = \mathrm{diag}[m_1^s, \ldots, m_p^s]$ and $D^s = \mathrm{diag}[d_1^s, \ldots, d_p^s]$ are equivalent inertia and damping matrices, respectively, $\tilde{u}(t)$ is the same disturbance input as in (5.1), and $L_m^s \in \mathbb{R}^{p\times p}$ is the equivalent asymmetric Laplacian matrix with elements

$$[L_m^s]_{ki} = \begin{cases} -a_{ki}^s/m_k^s & \text{if } k \neq i, \\ \sum_{\kappa\neq k} a_{k\kappa}^s/m_k^s & \text{if } k = i. \end{cases} \tag{5.4}$$

Notice that B^s in (5.3) now indicates that the input enters the lth area. The equivalent inertias m_k^s and damping factors d_k^s in (5.3) and the equivalent edge weights a_{ki}^s in (5.4) are unknown, and need to be estimated. We first assume that the equivalent topology $\mathcal{G}^s$ is a complete graph, meaning that every area is connected to every other area; then we set a threshold $\alpha \in \mathbb{R}$ such that if $a_{ki}^s < \alpha$ then $a_{ki}^s = 0$, i.e., areas k and i are not connected.

Because of the clustering and resulting time-scale separation of the states in (5.1), the phase angle of the ith machine can be written as

$$\Delta\delta_i(t) = \Delta\delta_k^s(t) + \Delta\delta_i^f(t), \tag{5.5}$$

for $i \in \mathcal{V}^k$, $k = 1, \ldots, p$, and where $\Delta\delta_i^f(t)$ denotes the fast component of $\Delta\delta_i(t)$ following from the fast eigenvalues of A in (5.1). The aggregate state $\delta_k^s(t)$ can be extracted from the measured values of $\Delta\delta_i(t)$, where $t = t_0 \ldots t_f$, using modal decomposition techniques such as Prony's method [Hau90] or the eigenvalue realization algorithm (ERA) [JP85] described in Appendix B.

Once $\Delta\delta^s(t)$ is estimated, the ISO estimates the equivalent state matrix A^s defined in (5.3) by formulating a nonlinear least-squares (NLS) problem as

$$\min_{A^s} \int_{t_0}^{t_f} \|\Delta\delta^s(t) - \Delta\hat{\delta}^s(t, A^s)\|_2^2 dt, \tag{5.6a}$$

$$\text{s.t.} \quad \Delta\hat{\delta}^s(t, A^s) = \hat{C} \exp(A^s t) \begin{bmatrix} \Delta\hat{\delta}^s(t_0) \\ \Delta\hat{\omega}^s(t_0) \end{bmatrix}, \tag{5.6b}$$

where $\hat{C} = [I_p,\ 0_p]$. The NLS problem (5.6) can be solved in real-time using standard libraries such as MATLAB's optimization toolbox. Once A^s is estimated, the ISO extracts L_m^s from its bottom left $p \times p$ block shown in (5.3), and constructs the equivalent topology by inspecting the elements of L_m^s. For completeness, the steps of this algorithm are summarized in Algorithm 3. The resulting L_m^s is the most up-to-date estimation of the equivalent topology, and therefore will provide more accurate localization compared to the results in [NC13] where the equivalent topology was assumed to be fixed throughout.

Algorithm 3 Online Topology Identification

Require: Measurements $y(t) = C\Delta\delta(t)$ for $t = t_0, \ldots, t_f$

1: Extract $\delta_k^s(t)$ for $k = 1, \ldots, p$
2: Estimate A^s from NLS problem (5.6)
3: Extract L_m^s from A^s

5.1.2 Step 2—Constructing Localization Keys

For continuity, we recapitulate the localization results from Chapter 3. We first assume that $\tilde{u}(t)$ is an impulsive function in the lth area and consider a single output of frequency—not phase angle—measurements from the kth equivalent generator (5.3). In the case of frequency measurement, the input-output transfer function is

$$g_{kl}^s(s) \approx \sum_{i=1}^{p} \frac{R_{kl}^i}{s^2 + \lambda_i}, \tag{5.7}$$

where λ_i is the ith eigenvalue of L_m^s. We use '$\approx$' in (5.7) because we have neglected any damping for simplicity. The residue R_{kl}^i in (5.7) can be written as the product of the mode controllability

factor and mode observability factor [ML89] as

$$R^i_{kl} = \underbrace{[v_i]_l}_{\text{obsv. factor}} \underbrace{[w_i^T M^{-1}]_k}_{\text{ctrl. factor}}, \tag{5.8}$$

where v_i and w_i are the right and left eigenvectors corresponding to λ_i of L^s_m. Hence, the sign of R^i_{kl} must be determined by the relative nodal domain locations of k and l. For example, if $\mathcal{D}_1$ and $\mathcal{D}_2$ are adjacent λ_i nodal domains (Definition 2.2.2), then

$$\operatorname{sgn}(R^i_{kl}) = \begin{cases} + & \text{if } k, l \in \mathcal{D}_1 \setminus \mathcal{D}_2, \\ - & \text{if } k \in \mathcal{D}_1 \setminus \mathcal{D}_2 \text{ and } l \in \mathcal{D}_2 \setminus \mathcal{D}_1, \\ 0 & \text{if } k \in \mathcal{D}_1 \cap \mathcal{D}_2 \text{ or } l \in \mathcal{D}_1 \cap \mathcal{D}_2. \end{cases}$$

It is straightforward to generalize this example to all of the λ_i nodal domains (not only adjacent nodal domains). Thus, using L^s_m identified in Algorithm 3, we can build a mapping between the relative input-output locations and the transfer function residues (5.8). We refer to this mapping as a *localization key*, and define it formally as

$$\mathcal{K}_{kl} = \{\operatorname{sgn}([v_2]_l[w_2]_k), \ldots, \operatorname{sgn}([v_p]_l[w_p]_k)\}. \tag{5.9}$$

We also define an array of these localization keys as $\mathcal{K} = [\mathcal{K}_{kl}]$, which provides a mapping of all input-output locations (at the area level). In other words, the kth row (column) of $\mathcal{K}$, denoted by $\mathcal{K}_k$, is associated with Area k. This row (column) consists of the keys that relate an input in any (other) area to an output taken from Area k. Furthermore, if we consider every input-output pair, we may define a matrix of residues corresponding to each eigenvalue as

$$R^i = [R^i_{kl}] = v_i w_i^T M^{-1} \tag{5.10}$$

for $i = 1, \ldots, p$. The matrix R^i in (5.10) is symmetric, and the λ_i nodal domains of v_i and w_i are identical, which implies that the array of localization keys $\mathcal{K}$ is symmetric [NC13].

Algorithm 4 outlines the steps which the central operator must take in order to construct $\mathcal{K}$ in real-time. The most complex step occurs in line 1, which involves an eigenvalue decomposition of the $p \times p$ matrix L^s_m. Starting on line 3, the individual keys $\mathcal{K}_{kl}$ are constructed according to (5.9). Notice that the array of keys is symmetric and the diagonal entries are trivial, i.e. $\mathcal{K}_{kk} = \{+, \ldots, +\}$. Hence, only the upper triangle of the array needs to be computed. Finally, the central operator transmits the row of keys $\mathcal{K}_k$ to the respective local area. In Section 5.2 we illustrate the construction of an array of localization keys for a four-area equivalent partitioning of the IEEE 39-bus system.

As we described in Chapter 4, the real component of the residues R^i_{kl} in (5.7) of the transfer

Algorithm 4 Centralized Localization Key Building for the Equivalent Network

Require: Equivalent Laplacian L_m^s

1: Compute eigenvalues and eigenvectors of L_m^s

2: **for** $k = 1 \to p$ **do**

3: Construct row of localization keys $\mathcal{K}_k$

$$[\mathcal{K}_k]_l = \{\text{sgn}([v_2]_l[w_2]_k), \ldots, \text{sgn}([v_p]_l[w_p]_k)\}$$

▷ Only require $l > k$ since keys are symmetric

4: **end for**

5: Broadcast localization keys to Area Operators

function from $\tilde{u}(t)$ to $\Delta\omega^s(t)$ contains information relating the location of the input relative to the location of the output through the λ_i nodal domains (Definition 2.2.1). We have shown that the residues of the transfer function from $\tilde{u}(t)$ in (5.3) to $y(t)$ in (5.2) contain identical information. In that case, however, the information is stored in the imaginary part of the transfer function residues. It follows that the operator in the ith area can easily use her measurements $y^i(t)$ to estimate the signs of the residues in (5.8) and exploit the information they carry. We describe this algorithm next.

5.1.3 Step 3—Real-time Input Localization

Step 3 is carried out entirely by the local area operators individually, i.e., each operator uses her own and only her own measurements. The goal of this step is to exploit the locational information carried by the transfer function residues (recall the discussion from the previous section). This can be understood in three distinct parts: estimate the system slow poles and residues of (5.7), build an estimated localization key $\hat{\mathcal{K}}_{i\kappa}$, then compare the estimated key to the set provided by the central operator. Algorithm 5 summarizes Step 3 and its three parts are forthwith described in detail.

5.1.3.1 Estimate slow poles and residues

The operator in Area i must estimate the equivalent transfer function (5.7), in particular, he must determine its poles and residues. It is important to emphasize that this part of Step 3—lines 1–2 in Algorithm 5—can be carried out in parallel to Step 1, i.e., starting at $t = t_f$, which is being executed by the central ISO. Since we have assumed $\mathcal{G}^s$ is connected, the operator in Area i can use her measurements $y^i(t) = C_i\Delta\delta$ from a set of generator buses $\mathcal{S}^i \subset \mathcal{V}^i$ to find the poles and residues of interest. This is done by running a standard system identification algorithm to construct the transfer function from $\tilde{u}(t)$ in (5.1) to $y^i(t)$, then truncating this transfer function

to only consider the slow poles and residues. The result of this subroutine is the estimated values of the system poles and residues, which we denote by $\hat{\lambda}_k$ and $\hat{R}^k_{i\kappa}$, respectively, for $k = 1, \ldots, \hat{p}$. Notice that since the input location is unknown we have used the dummy variable κ. Also notice that depending on the type of disturbance, e.g. transient, harmonic, etc., and due to other un-modeled dynamics in the power system, the system identification algorithm may identify more than $p - 1$ slow oscillatory components. Hence, we use $\hat{p}$ to denote the total number of slow modes (including the dc mode) recovered by the system identification.

5.1.3.2 Build estimated localization key $\hat{\mathcal{K}}_{i\kappa}$

Before proceeding further, the local operator requires the modes of A^s and localization keys $\mathcal{K}_i$. The slow oscillatory components closest to the natural modes of A^s and their corresponding residues are used to build an estimated localization key $\hat{\mathcal{K}}_{\kappa i}$ according to (5.8) and (5.9). In the case that additional slow modes are identified, the residues used to build $\hat{\mathcal{K}}_{\kappa i}$ may need to be corrected.

5.1.3.3 Compare estimated key

Recall that the signs of the transfer function residues contain information about the relative input-output nodal domain locations, and that the estimated $\hat{\mathcal{K}}_{\kappa i}$ is a mapping between the residues and these locations. Since the output location i is known, the localization problem can be solved by determining κ. This is done by comparing $\hat{\mathcal{K}}_{\kappa i}$ to the most up-to-date set of keys $\mathcal{K}_i$ provided by the central ISO. Matching key(s) indicate possible area(s) in which the disturbance input may have entered. For example, if $\hat{\mathcal{K}}_{\kappa i} = [\mathcal{K}_i]_3$, then the disturbance may have occurred in Area 3. Notice that this matching does not have to be unique. In fact, it should be clear that this matching will only be unique if for all $k \neq l$ we have $[\mathcal{K}_i]_k \neq [\mathcal{K}_i]_l$ (cf. Theorem 3.2.2). For example, if the equivalent network is a complete graph, then the set of keys $\mathcal{K}_i$ constructed in Step 2 will preclude uniquely identifying the area in which the disturbance entered. We refer the reader to Chapter 3 for further discussion of classes of graphs in which this step will result in a unique localization. We next provide a concrete implementation of Steps 1–3 using a realistic power system model.

5.2 Case-study—IEEE 39-bus network

This case study provides a concrete implementation of Steps 1–3 following a disturbance in a modified IEEE 39-bus model[1], shown in Fig. 5.2. The model parameters of Fig. 5.2 have been derived from [Ath79]. We consider a 4-area partitioning of the network.

[1]The line typically connecting buses 16 and 17 has been removed to improve the coherency of the areas.

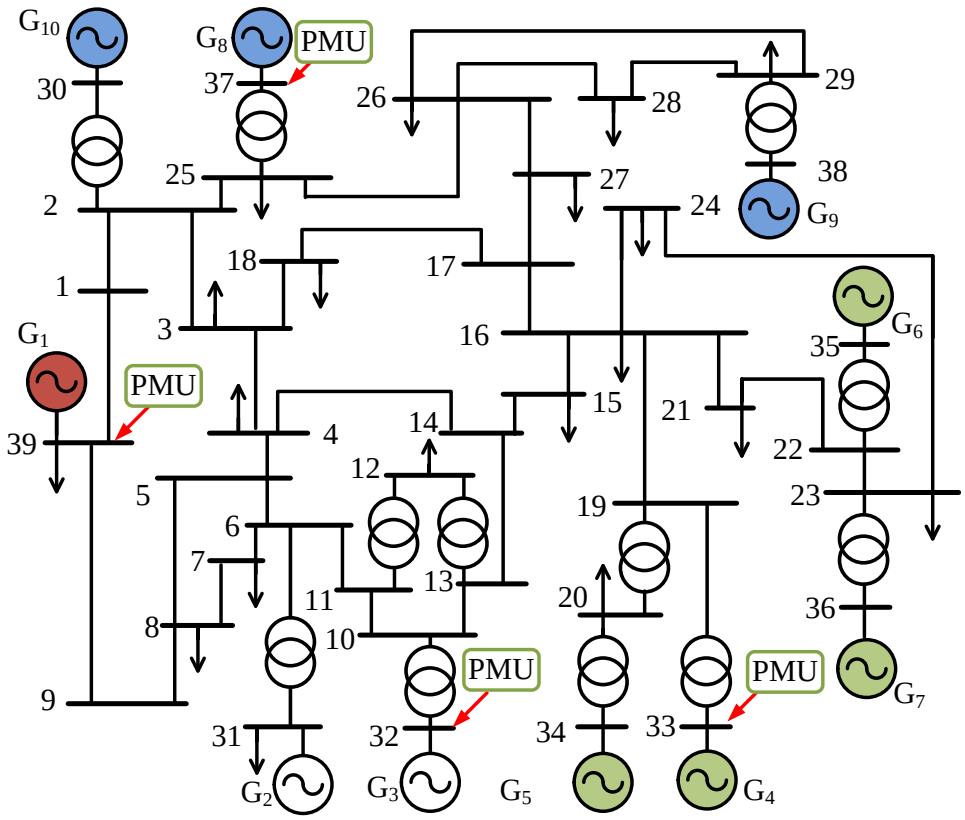


Figure 5.2 (a) IEEE 39-bus power system model used in the case study, with areas: $G_1^s = \{G_1\}$, $G_2^s = \{G_2,\ G_3\}$, $G_3^s = \{G_4,\ G_5,\ G_6,\ G_7\}$, $G_4^s = \{G_8,\ G_9,\ G_{10}\}$.

Algorithm 5 Real-time attack localization from Area i

Require: Local PMU measurements, localization keys $\mathcal{K}_i$, modes of A^s

1: Run System Identification to estimate $(\hat{A}^s, \hat{\tilde{B}}^s, \hat{C}^s)$
2: Extract slow modes $\hat{\lambda}_k$ $k = 2, \ldots, \hat{p}$ and residues $\hat{R}^k_{\kappa i}$, $k = 2, \ldots, \hat{p}$,
3: Wait to receive $\mathcal{K}_i$ and modes of L^s_m from central operator
4: Construct estimated localization key

$$\hat{\mathcal{K}}_{\kappa i} = \{\text{sgn}(\hat{R}^2_{\kappa i}), \ldots, \text{sgn}(\hat{R}^p_{\kappa i})\}$$

5: **for** $\kappa = 1 \to p$ **do**
6: Compare $\hat{\mathcal{K}}_{\kappa i}$to elements of $\mathcal{K}_i$
7: **if** Match, i.e., $\hat{\mathcal{K}}_{\kappa i} = [\mathcal{K}_i]_l$ **then**
8: Attack may have occurred in Area l
9: **end if**
10: **end for**

At $t_0 = 0$ a three-phase fault occurs on the line between buses 3 and 4, which can be considered as part of Area 4. By $t_f = 6$ sec enough data has been recorded to capture the slow oscillatory modes of the network. Fig. 5.3 shows the output measurements that are sent to the central operator. Steps 1–3 are carried out as follows.

5.2.1 Step 1

At $t = t_f$, the central operator takes the measurements $y(t)$ shown in Fig. 5.3 and starts Algorithm 3. Next, the NLS problem (5.6) is formulated and solved in t_1 seconds. The identified equivalent topology $\mathcal{G}^s$ is shown in Fig. 5.4(a), and the asymmetric Laplacian matrix is

$$L^s_m = \begin{bmatrix} -8.883 & 5.497 & 0 & 3.386 \\ 12.828 & -33.159 & 16.981 & 3.350 \\ 0 & 8.612 & -11.174 & 2.562 \\ 10.392 & 4.406 & 6.645 & -21.443 \end{bmatrix}. \tag{5.11}$$

Notice that in Fig. 5.4(a), Area 1 and Area 3 are not connected and the corresponding elements in the identified Laplacian matrix (5.11) are identically zero. The eigenvalues of A^s are

$$\{-0.30 \pm 3.083j,\ -0.27 \pm 4.966j,\ -0.28 \pm 6.297j\}. \tag{5.12}$$

5.2.2 Step 2

At $t = t_f + t_1$, the central operator next runs Algorithm 4 to determine the array of localization keys. He first runs an eigenvalue decomposition on L^s_m in (5.11) to determine the λ_k nodal

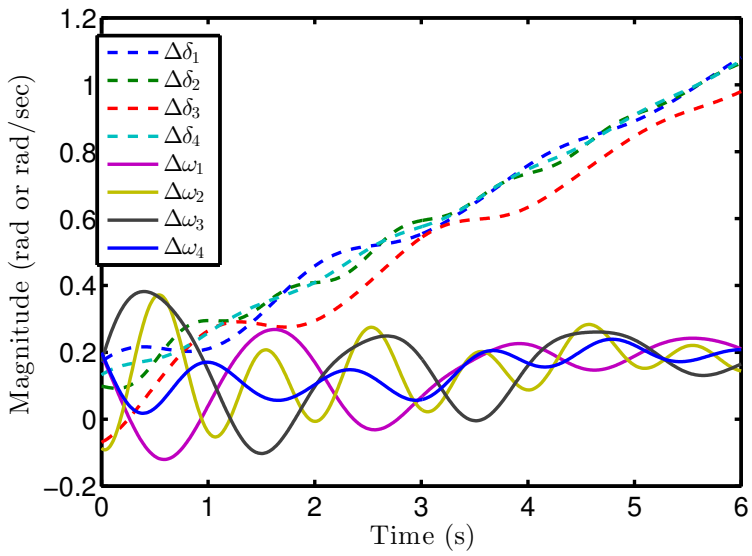


Figure 5.3 Measured output following a fault in the IEEE 39-bus model.

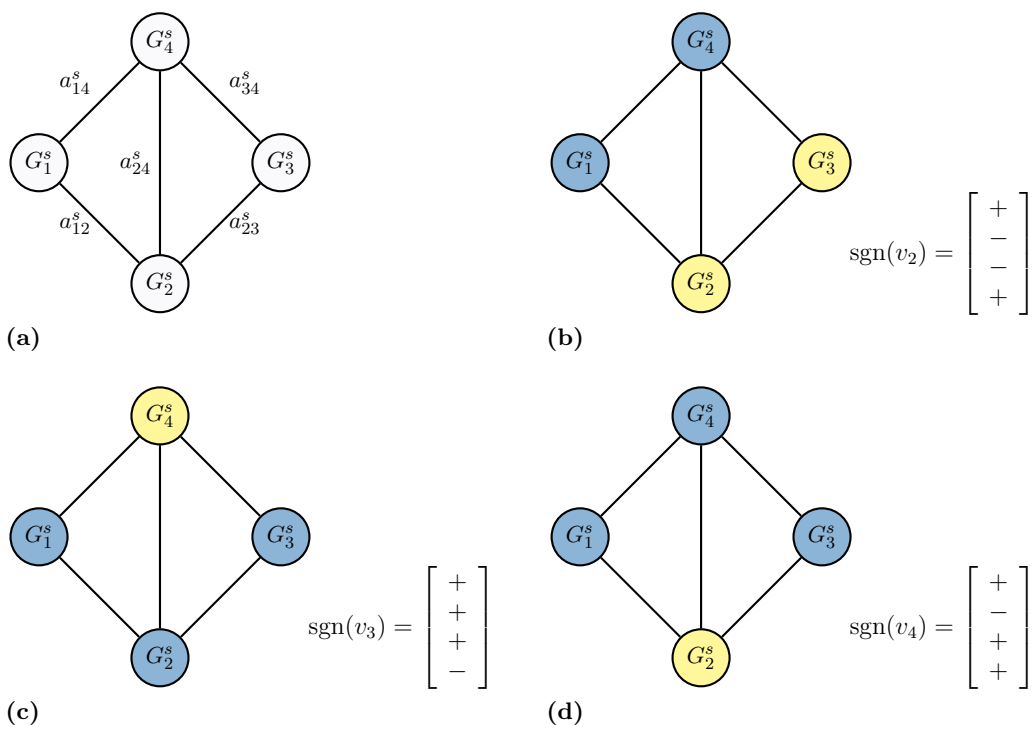


Figure 5.4 (a) Identified $\mathcal{G}^s$ for IEEE 39-bus model; (b)–(d) λ_2–λ_4 nodal domains of $\mathcal{G}^s$. Notice that there are only two nodal domains for each of λ_2–λ_4.

Table 5.1 Input localization keys for the IEEE 39-bus equivalent system in Fig. 5.2

Area	1	2	3	4
1	$\cdot$	$\{-,+,-\}$	$\{-,+,+\}$	$\{+,-,+\}$
2	$\cdot$	$\cdot$	$\{+,+,-\}$	$\{-,-,-\}$
3	$\cdot$	$\cdot$	$\cdot$	$\{-,-,+\}$
4	$\cdot$	$\cdot$	$\cdot$	$\cdot$

domains of $\mathcal{G}^s$ for $k = 2, \ldots, 4$. The resulting nodal domains and eigenvector sign patterns are shown in Fig. 5.4. The array of localization keys $\mathcal{K}$ constructed using these nodal domains is provided in Table 5.1. This entire step takes t_2 seconds. The central operator transmits individual rows of Table 5.1 to the local area operator corresponding to that row, e.g., the first row is sent to Area 1. Assuming there is some network delay t_d, $\mathcal{K}_k$ is received at control room of Area k at time $t = t_f + t_1 + t_2 + t_d$.

5.2.3 Step 3

The local area operators can begin Step 3 at the same time as the central operator begins Step 1. For this case study, we consider the perspective of the operator sitting in Area 1. Using her measurements $y^1(t),\ t = t_0, \ldots, t_f$ from generator 1, he starts Algorithm 5 by running a system identification algorithm such as ERA. The estimated poles and residues are shown in Table 5.2. Notice that two additional slow poles were identified by ERA. If this subroutine takes $t_3 < t_1 + t_2 + t_d$ seconds to execute, the operator in Area 1 must wait until he receives $\mathcal{K}_1$ (first row of Table 5.1) and the system slow poles from the central ISO at $t = t_f + t_2 + t_d$ before proceeding.

Once the set of keys $\mathcal{K}_1$ and network slow poles (5.12) have been received, uses only her estimated poles in Table 5.2 that match the ones sent by the central ISO (shown in bold), and uses the corresponding residues to construct $\hat{\mathcal{K}}_{\kappa 1}$. Evaluating the sign of the real part of the residues in Table 5.2 according to (5.9), he finds $\hat{\mathcal{K}}_{\kappa 1} = \{+,-,+\}$. Finally, the operator in Area 1 finished executing Algorithm 5 by comparing her estimated key $\hat{\mathcal{K}}_{\kappa 1}$ to $\mathcal{K}_1$ (first row of Table 5.1). Clearly there is a match in the fourth column, i.e., $\hat{\mathcal{K}}_{\kappa 1} = [\mathcal{K}_1]_4$. The operator in Area 1, therefore, infers that the disturbance has occurred in Area 4. The same procedure can be repeated by the operators in Areas 2, 3, and 4 as well.

Table 5.2 Poles and residues recovered from $\Delta\omega_1$. Values that are used to build the localization key are in **boldface**.

Pole	Residue
$-0.3447 + 0.9173i$	$0.0299 + 0.0794i$
$-0.3444 + 2.0210i$	$-0.0468 + 0.0097i$
$\mathbf{-0.3011 + 3.0825i}$	$\mathbf{0.0327 + 0.1115i}$
$\mathbf{-0.2634 + 4.9840i}$	$\mathbf{-0.0044 - 0.0121i}$
$\mathbf{-0.2969 + 6.2634i}$	$\mathbf{0.0112 - 0.0027i}$

Part II

Control

Chapter 6

Optimal Control of Graphs with Nodal Domain Constraints

For the next two chapters, we consider a network of linear consensus agents defined over the connected graph $\mathcal{H} = (\mathcal{V},\ \mathcal{E})$

$$\dot{x}(t) = -L_{\mathcal{H}}x(t) + Bu(t) + \tilde{B}\tilde{u}(t) \tag{6.1a}$$

$$y(t) = Cx(t), \tag{6.1b}$$

where the states $x(t) \in \mathbb{R}^n$, control inputs $u(t) \in \mathbb{R}^m$, disturbance inputs $\tilde{u} \in \mathbb{R}^{\tilde{m}}$, outputs $y(t) \in \mathbb{R}^h$, and state matrices are all defined as in (2.13). $L_{\mathcal{H}} \in \mathcal{L}_{\mathcal{H}}^{+}$ is the graph Laplacian matrix of the underlying network $\mathcal{H}$. We assume, without loss of generality, that $M = I_n$. We also assume the graph can be partitioned into p coherent areas as $\mathcal{V} = \bigcup_{k=1}^{p} \mathcal{V}^k,\ \mathcal{V}^i \cap \mathcal{V}^j = \emptyset,\ i \neq j$, where the number of nodes in the kth area is denoted $n_k = |\mathcal{V}^k|$. The kth area is the induced subgraph $\mathcal{H}^k = \mathcal{H}[\mathcal{V}^k]$. As in Section 2.4, we define the slow states $x_s = [x_s^1, \ldots, x_s^p]^T$ as the average of the states within an area. For example, the slow state corresponding to the kth area is defined as

$$x_s^k(t) = \frac{1}{n_k} \sum_{i \in \mathcal{V}^k} x_i(t). \tag{6.2}$$

To recall, fast states are defined for each area by comparing the state relative to a reference within the same area. For example,

$$x_f^{k,i}(t) = x_i - x_{i^*}, \quad \forall i \neq i^*,\ i, i^* \in \mathcal{V}^k, \tag{6.3}$$

where $i^* \in \mathcal{V}^k$ is the index of the reference agent. Since the fast states converge to a pseudo-steady state relatively quickly, we are primarily interested in studying the aggregate system with states defined by (6.2), which can be approximated by a rigid body aggregate network [CK85] over the graph $\mathcal{G}$ with perturbed dynamics described by

$$\hat{M}\dot{x}_s(t) = -L_{\mathcal{G}}x_s(t) + \bar{u}_s(t) + \mathbf{e}_k\tilde{u}(t), \tag{6.4}$$

where $L_{\mathcal{G}} = EWE^T \in \mathcal{L}_{\mathcal{G}}^+$ is the graph Laplacian matrix of the aggregate network, $\bar{u}_s(t) \in \mathbb{R}^p$ is the aggregate control input which can be found by applying the transformation (2.19) on $u(t)$, and $\mathbf{e}_k\tilde{u}(t)$ represents a disturbance entering the kth area. The details of how to implement the aggregate control input $\bar{u}_s(t)$ are developed in Chapter 7.

For now, recall that in Part I we developed graph-theoretic methods to localize the cluster in which a disturbance input $\tilde{u}(t)$ (ignoring the control inputs). In this case, the transfer function (TF) from input $\tilde{u}(t)$ in area i to output $y_j(t) = \mathbf{e}_j x_i(t)$ can be written as

$$g_{ij}^{slow} = \sum_{k=1}^{p} \frac{R_{ij}^k}{s + \lambda_k}. \tag{6.5}$$

We showed that the disturbance $\tilde{u}(t)$ can be localized up to the cluster level by analyzing $\text{sgn}(R_{ij}^k)$ for $k = 2, \ldots, p$. In Part II, it is our objective to shape the modes, that is, both the eigenvalues and *eigenvectors* of the slow subsystem defined by the aggregate states (6.2) through feedback control. Specifically, the observability and controllability of a particular mode (e.g. the slow modes) of the network can be characterized in terms of the nodal domains or set of zero nodes (cf. Theorem 3.2.1). Recall that a necessary condition for the localization algorithm (Algorithm 2) was that the transfer function (6.5) is minimal. To guarantee that this condition is met, we introduce an aggregate feedback control input with the form

$$\bar{u}_s(t) = -EKE^T x_s(t), \tag{6.6}$$

where $E \in \mathbb{R}^{n\times m}$ is the incidence matrix of graph $\mathcal{G}$ and $K = \text{diag}[k_{e_1}, \ldots, k_{e_m}] \in \mathbb{R}^{m\times m}$ with $e_k \in \mathcal{E}$. We will let $L_{\mathcal{K}} = EKE^T$ denote the graph Laplacian of the aggregate control graph $\mathcal{K}$. By designing our feedback in this way, we ensure that the feedback conforms to a specified structure. Since $\mathcal{K}$ can be thought of as a cyber or communication enabled graph, we restrict its structure to be a subset of the physical graph. In other words, the only admissible communication links correspond to preexisting physical edges. Note that unlike the physical graph, the control graph may have negative edge weights. However, in order that the problem retains physical meaning, we require that the closed loop graph have positive edge-weights, i.e. $L_{cl} = L_{\mathcal{G}} + L_{\mathcal{K}} \in \mathcal{L}_{\mathcal{G}}^+$. Therefore, the problem can be stated as:

Optimally re-weight the edges of $\mathcal{G}$ via $\mathcal{K}$ such that $\tilde{u}(t)$ is guaranteed to be localizable in the closed-loop system (A.12).

We can formalize this problem by defining $\mathcal{C}_k^*$ as the set of feasible nodal domain configurations with no zero nodes for mode λ_k. For example, consider the three node path in 6.1(a). There are three feasible nodal domain configurations for λ_2 (Fig. 6.1(b)–6.1(d)):

$$\mathcal{C}_2 = \left\{ \begin{bmatrix} + \\ + \\ - \end{bmatrix}, \begin{bmatrix} + \\ 0 \\ - \end{bmatrix}, \begin{bmatrix} + \\ - \\ - \end{bmatrix} \right\}, \tag{6.7}$$

one of which contains a zero node. Hence,

$$\mathcal{C}_2^* = \left\{ \begin{bmatrix} + \\ + \\ - \end{bmatrix}, \begin{bmatrix} + \\ - \\ - \end{bmatrix} \right\}. \tag{6.8}$$

The design problem can then be formalized as: for each $k = 2, \ldots, p$ choose $\psi_k \in \mathcal{C}_k^*$, then find K such that $W + K \in \mathcal{W}_\psi$, where

$$\mathcal{W}_\psi = \cap \mathcal{W}_{\psi_k}, \tag{6.9}$$

$$\mathcal{W}_{\psi_k} = \{W = \operatorname{diag}[w_1, \ldots, w_n] \succ 0_n \mid \operatorname{sgn}(v_k) = \psi_k\}, \tag{6.10}$$

with $L_{\mathcal{G}} v_k = \lambda_k v_k$. For this chapter we assume that the set $\{\psi_k\}$ is chosen such that $\mathcal{W}_\psi \neq \emptyset$. Given this assumption we would like to choose the K with $W + K \in \mathcal{W}_\psi$ that maximizes (or minimizes) a closed-loop performance objective $J(K)$. Formally,

$$\begin{aligned} \underset{K}{\text{maximize}} \quad & J(K) && \text{(6.11a)} \\ \text{subject to} \quad & E(W+K)E^T \in \mathcal{L}_{\mathcal{G}}^+, && \text{(6.11b)} \\ & W + K \in \mathcal{W}_\psi, && \text{(6.11c)} \\ & aI_m \succ K \succ bI_m;\ K \text{ diagonal} && \text{(6.11d)} \end{aligned}$$

where $a,\ b \in \mathbb{R}$ are upper and lower bounds on the magnitude of the gains. Notice that the lower bound b must satisfy

$$b > -\max_{e_k}(w_{e_k}), e_k \in \mathcal{E}. \tag{6.12}$$

Examples of performance objectives $J(K)$ include minimizing the $\mathcal{H}_2/\mathcal{H}_\infty$ gains of the closed loop transfer function [Lin11; Far14; Sch12], maximizing consensus speed, and bounding certain

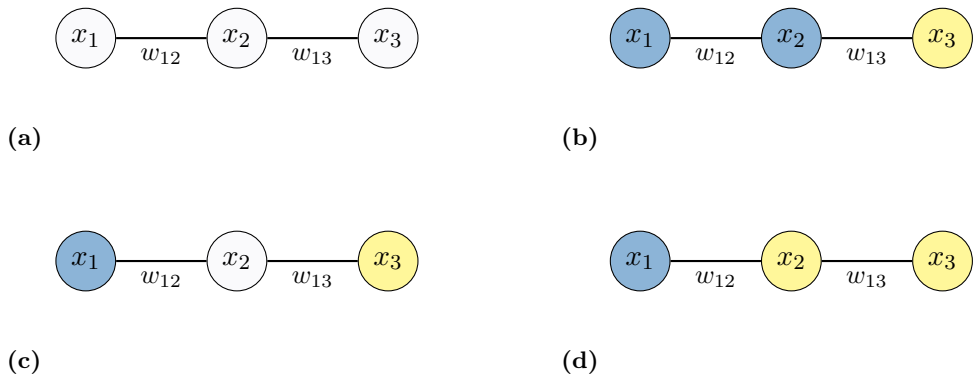


Figure 6.1 (a) Three-node path graph $\mathcal{P}_3$; (b)–(d) λ_2 nodal domain configurations.

closed-loop eigenvalues [Sha12a], to name a few. Additionally, sparsity-promoting terms can easily be incorporated into the objective. Before concretely implementing our control design, we note that the above problem (6.11) can be reformulated as a closed-loop graph design problem with $\tilde{W} = W + K$, as

$$\begin{aligned} \underset{\tilde{W}}{\text{maximize}} \quad & J(\tilde{W}) && \text{(6.13a)} \\ \text{subject to} \quad & E\tilde{W}E^T \in \mathcal{L}_{\mathcal{G}}^{+}; && \text{(6.13b)} \\ & \tilde{W} \in \mathcal{W}_{\psi}; && \text{(6.13c)} \\ & \tilde{a}I_m \succ \tilde{W}. && \text{(6.13d)} \end{aligned}$$

The remainder of this chapter is devoted to developing the theoretical underpinnings of problem (6.13) while restricting our attention to choosing $\psi_2 \in \mathcal{C}_2^*$ on tree graphs. In the following chapter we will show how to concretely implement this control strategy in large-scale clustered networks.

6.1 λ_2 Nodal Domains on Trees

The constraints in (6.13) are not convex, in general. To understand this better, we recall the following characterization of the λ_2 nodal domains of trees.

6.1.1 Characterization of λ_2 Nodal Domains on Trees

The following theorems (due to Kirkland et al.) characterize the λ_2 nodal domains of weighted trees. These theorems will in turn be used to generate the set of edge-weight constraints for any feasible nodal domain configuration on a weighted tree. The first theorem pertains to trees with no zero vertices, referred to as *Type II* trees.

In Type II trees, adjacent vertices in neighboring nodal domains are called *characteristic vertices*. The name is appropriate because these two adjacent vertices completely characterize the λ_2 nodal domains of Type II trees.

Remark 6.1.1. *Every $\psi_2 \in \mathcal{C}_2^*$ has a one to one correspondence with an edge $ij \in \mathcal{E}$, where $i \sim j$ are characteristic vertices.*

This means that specifying the design choice $\psi_2 \in \mathcal{C}_2^*$ is equivalent to specifying the characteristic vertices $i \sim j \in \mathcal{V}$.

Theorem 6.1.1 (Type II weighted trees [Kir96])**.** *Let $\mathcal{T}$ be a weighted tree on n vertices $1, \ldots, n$ with Laplacian matrix $L_{\mathcal{T}}$. Suppose that i and j are adjacent vertices of $\mathcal{T}$. Then $\mathcal{T}$ is Type II with characteristic vertices i and j if and only if there exists $0 < \gamma < 1$ such that*

$$\rho(P_{ij} - \frac{\gamma}{w_{ij}}\mathbf{J}) = \rho(P_{ji} - \frac{1-\gamma}{w_{ij}}\mathbf{J}), \tag{6.14}$$

where P_{ij} is the bottleneck matrix of the branch at i containing vertex j and vice-versa, $\rho(\cdot)$ is the spectral radius operator, and $\mathbf{J}$ is of conformable size. Moreover, if (6.14) holds then

$$\frac{1}{\lambda_2} = \rho(P_{ij} - \frac{\gamma}{w_{ij}}\mathbf{J}) = \rho(P_{ji} - \frac{1-\gamma}{w_{ij}}\mathbf{J}). \tag{6.15}$$

The bottleneck matrices are related to the edge weights through the inverse of the principal submatrix of $L_{\mathcal{G}}$ grounded at vertex k, denoted L_k. From [Kir96] (Proposition 1) we have

$$[L_k^{-1}]_{ij} = \sum_{e \in \mathcal{P}_{ij}^k} \frac{1}{w_e}, \tag{6.16}$$

where $\mathcal{P}_{ij}^k$ is the set of edges of $\mathcal{T}$ which are on both the path from i to k and from j to k. The bottleneck matrices make up the block diagonal $L_k^{-1} = \text{diag}[P_{k1}, \ldots, P_{kr}]$. The number of bottleneck matrices for a given L_k is equivalent to the number of connected subgraphs induced on the vertex set $\mathcal{V} \setminus k$. The bottleneck matrix P_{ki} can be thought of in terms of the graph Laplacian of these subgraphs. Concretely, we write the bottleneck matrices as

$$P_{ki}^{-1} = L_k[\mathcal{V}^i], \tag{6.17}$$

where $\mathcal{V}^i$ denotes the vertex set of the subgraph that contains node i, i.e. $i \in \mathcal{V}^i$, and L_k indicates the fact that the subgraph is grounded at node $k \in \mathcal{V}$.

The second theorem pertains to weighted trees where there is at least one zero node, referred to as *Type I* trees. In Type I trees, the characteristic vertex is the unique zero node with at least one non-zero neighbor.

Theorem 6.1.2 (Type I weighted trees [Kir96])**.** *Let $\mathcal{T}$ be a weighted tree on n vertices $\{1, \ldots, n\}$. Then $\mathcal{T}$ is a Type I tree with characteristic vertex k if and only if there are least two Perron branches of $\mathcal{T}$ at k. Moreover, the algebraic connectivity of $\mathcal{T}$ is $\lambda_2 = 1/\rho(L_k^{-1})$.*

Theorem 6.1.2 will be used for analysis purposes in Section 6.3.

6.1.2 Nonlinear Optimization Problem

Theorem 6.1.1 immediately gives us a set of equality constraints to use in our optimal control formulation (6.13). In particular, for $\psi = \psi_2 \in \mathcal{C}_2^*$ characterized by $i, j \in \mathcal{V}$, we can replace the constraint (6.13c) by

$$\rho(P_{ij} - \frac{\gamma}{\tilde{w}_{ij}}\mathbf{J}) = \rho(P_{ji} - \frac{1-\gamma}{\tilde{w}_{ij}}\mathbf{J}), \tag{6.18}$$

where $0 < \gamma < 1$, P_{ij} is defined in (6.17), and $\mathbf{J}$ is of conformable size. Because (6.18) represents a nonlinear (and nonconvex) transformation of the weight matrix $\tilde{W}$, we will approach numerical simulations and nonlinear programming (NLP) toolboxes such as Matlab's `fmincon` and `gaopt`. We next describe in detail how the constraint (6.18) can be incorporated into two important optimal graph design problems—namely, the fastest consensus problem and the minimum $\mathcal{H}_2$ norm problem.

$\mathcal{H}_2$-Norm Minimization

The $\mathcal{H}_2$ transfer function norm characterizes how noise propagates through the system. In other words, it indicates the total energy dissipated by the transfer function. To quantify the performance, we define the performance output for (6.1a) as

$$z(t) = \begin{bmatrix} Q^{1/2} & 0_m \\ 0_n & R^{1/2} \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}, \tag{6.19}$$

where $Q \succ 0_n$ and $R \succ 0_m$. The optimal control objective can be written as

$$\begin{aligned} J(\tilde{W}) &= \underset{\tilde{W}}{\text{minimize}} \ \|G_{cl}(s)\|_{\mathcal{H}_2}^2 \\ &= \underset{\tilde{W}}{\text{minimize}} \ \text{tr}\left[\tilde{B}^T \int_0^\infty e^{-\tilde{L}^T t}\left((Q + (\tilde{L} - L_{\mathcal{G}})^T R(\tilde{L} - L_{\mathcal{G}})\right) e^{-\tilde{L}t} dt \tilde{B}\right]. \end{aligned} \tag{6.20}$$

where $G_{cl}(s)$ is the closed loop transfer function from the disturbances $\tilde{u}(t)$ to the performance output $z(t)$, and where $\tilde{L} = E\tilde{W}E^T \in \mathcal{L}_{\mathcal{G}}^+$. If the closed loop system is stable, the optimal control objective (6.20) can be written as

$$J(\tilde{W}) = \underset{\tilde{W}}{\text{minimize}} \ \text{tr}\left[\tilde{B}^T \mathbf{P} \tilde{B}\right], \tag{6.21}$$

where $\mathbf{P} = \mathbf{P}(\tilde{W}) \in \mathbb{R}^{n \times n}$ is the closed-loop controllability Grammian which satisfies the Lyapunov equation

$$\tilde{L}^T \mathbf{P} + \mathbf{P}\tilde{L} = Q + (\tilde{L} - L_{\mathcal{G}})^T R(\tilde{L} - L_{\mathcal{G}}). \tag{6.22}$$

We can interpret the object (6.21) as attempting to improve the performance of the re-weighted network in the sense of the $\mathcal{H}_2$-norm, while minimally reweighing the edges. We can enforce our desired nodal domain configuration by including the constraint (6.18). The constrained $\mathcal{H}_2$ optimal control problem for tree graphs becomes

$$\begin{aligned} \underset{\tilde{W},\mathbf{P},\gamma}{\text{minimize}} \quad & \text{tr}\left[\tilde{B}^T \mathbf{P} \tilde{B}\right] && \text{(6.23a)} \\ \text{subject to} \quad & E\tilde{W}E\mathbf{P} + \mathbf{P}E\tilde{W}E = Q + E(\tilde{W} - W)E^T RE(\tilde{W} - W)E^T && \text{(6.23b)} \\ & \rho(P_{ij} - \frac{\gamma}{\tilde{w}_{ij}}\mathbf{J}) = \rho(P_{ji} - \frac{1-\gamma}{\tilde{w}_{ij}}\mathbf{J}), && \text{(6.23c)} \\ & 1 > \gamma > 0, && \text{(6.23d)} \\ & \tilde{a}I_m \succ \tilde{W} \succ 0_m; \ \tilde{W} \text{ diagonal.} && \text{(6.23e)} \end{aligned}$$

Fastest Consensus

It is well-known that the consensus speed of a NDS is bounded by the second smallest eigenvalue of the underlying graph Laplacian [ME10]. Hence, in order to maximize the consensus of the network, it is sufficient to maximize the algebraic connectivity of the closed-loop graph, as

$$J(\tilde{W}) = \underset{\tilde{W}}{\text{maximize}} \lambda_2(\tilde{L}), \tag{6.24}$$

where $\tilde{L} = E\tilde{W}E^T \in \mathcal{L}_{\mathcal{G}}^+$, with the diagonal edge weight matrix $\tilde{a}I_m \succ \tilde{W} \succ 0_m$. The problem (6.24) can be solved using semidefinite programming (SDP) by introducing a scalar κ to bound $\lambda_2(\tilde{L})$, in which case we can state (6.24) as

$$\begin{aligned} \underset{\tilde{W},\kappa}{\text{maximize}} \quad & \kappa, && (6.25a) \\ \text{subject to} \quad & E\tilde{W}E^T - \kappa I_n \succeq 0_n, && (6.25b) \\ & \tilde{a}I_m \succ \tilde{W} \succ 0_m;\ \tilde{W} \text{ diagonal.} && (6.25c) \end{aligned}$$

In light of (6.18) and (6.25), our constrained problem (6.13) with the fastest consensus objective can be written as

$$\begin{aligned} \underset{\tilde{W},\kappa,\gamma}{\text{maximize}} \quad & \kappa && (6.26a) \\ \text{subject to} \quad & E\tilde{W}E^T - \kappa I_n \succeq 0, && (6.26b) \\ & \rho(P_{ij} - \frac{\gamma}{\tilde{w}_{ij}}\mathbf{J}) = \rho(P_{ji} - \frac{1-\gamma}{\tilde{w}_{ij}}\mathbf{J}), && (6.26c) \\ & 1 > \gamma > 0, && (6.26d) \\ & \tilde{a}I_m \succ \tilde{W} \succ 0_m;\ \tilde{W} \text{ diagonal.} && (6.26e) \end{aligned}$$

Problem (6.26) aims to maximize the consensus speed while enforcing our desired nodal domains. Over the next two sections we explore special cases of (6.26) because the results are intuitive and nicely visualized.

6.2 Four-node Trees

In this section we maximize the closed-loop consensus speed while meeting our localizability constraints for the aggregate network, which we assume is a connected four-node tree, by solving problem (6.26).

There are two distinct four-node trees, namely the four node path graph, denoted $\mathcal{P}_4$ (Fig. 6.2(a)), and the four-node star graph, likewise denoted $\mathcal{S}_4$ (Fig. 6.3(a)). While the problem (6.26) is non-convex for four-node trees, we can visualize the constraints using a three dimensional plot (one dimension for each edge). In each case, the edge weights are discretized over a three-dimensional lattice from 0.5–5.0 with a stepsize of 0.1. Next, a line search for γ from 0 to 1 is used to check conditions (6.26c) and (6.26d) for each point in the lattice. If both conditions are met, then the point is feasible. We can then compare the results of this 'brute force' approach

to the results using the NLP toolbox.

6.2.1 Four-node Path Graph $\mathcal{P}_4$

The four-node path graph has five feasible nodal domain configurations for λ_2 and three configurations with no zero nodes (i.e. that correspond to Type II trees):

$$\mathcal{C}_2 = \left\{ \begin{bmatrix} + \\ + \\ + \\ - \end{bmatrix}, \begin{bmatrix} + \\ + \\ 0 \\ - \end{bmatrix}, \begin{bmatrix} + \\ + \\ - \\ - \end{bmatrix}, \begin{bmatrix} + \\ 0 \\ - \\ - \end{bmatrix}, \begin{bmatrix} + \\ - \\ - \\ - \end{bmatrix} \right\}, \tag{6.27a}$$

$$\mathcal{C}_2^* = \left\{ \begin{bmatrix} + \\ + \\ + \\ - \end{bmatrix}, \begin{bmatrix} + \\ + \\ - \\ - \end{bmatrix}, \begin{bmatrix} + \\ - \\ - \\ - \end{bmatrix} \right\}. \tag{6.27b}$$

To illustrate this numerical search process, we consider the design choice $\psi_2 = [+\ +\ +\ -]^T$ which is shown in Fig. 6.2(b). As previously mentioned, the edge-weight space is discretized into a three-dimensional lattice from 0.5 to 5.0. The feasible points are plotted in Fig. 6.2(c) with an intensity corresponding to the normalized objective function. A larger objective value corresponds to a more intense color, with the maximal value shown in a slightly different hue. If the problem were convex we would expect to see a precise and regular pattern of feasible points inside of the lattice and a steady color gradient. Instead, we can see a number missing points or holes in Fig. 6.2(c). While the general trend of the color is from less intense to more intense, there are a number of bright spots which correspond to local optima of the objective function. The optimal value found for λ_2 using the brute force method is $\lambda_2 = 1.35$, which corresponds to edge weights $\tilde{W}^* = \text{diag}[4.8\ 4.9\ 1.2]^T$. Table 6.1 summarizes these results alongside the optimal value found using a the NLP toolbox and the nominal (unconstrained) solution[1].

[1]The unconstrained problem is convex, hence the solution is unique.

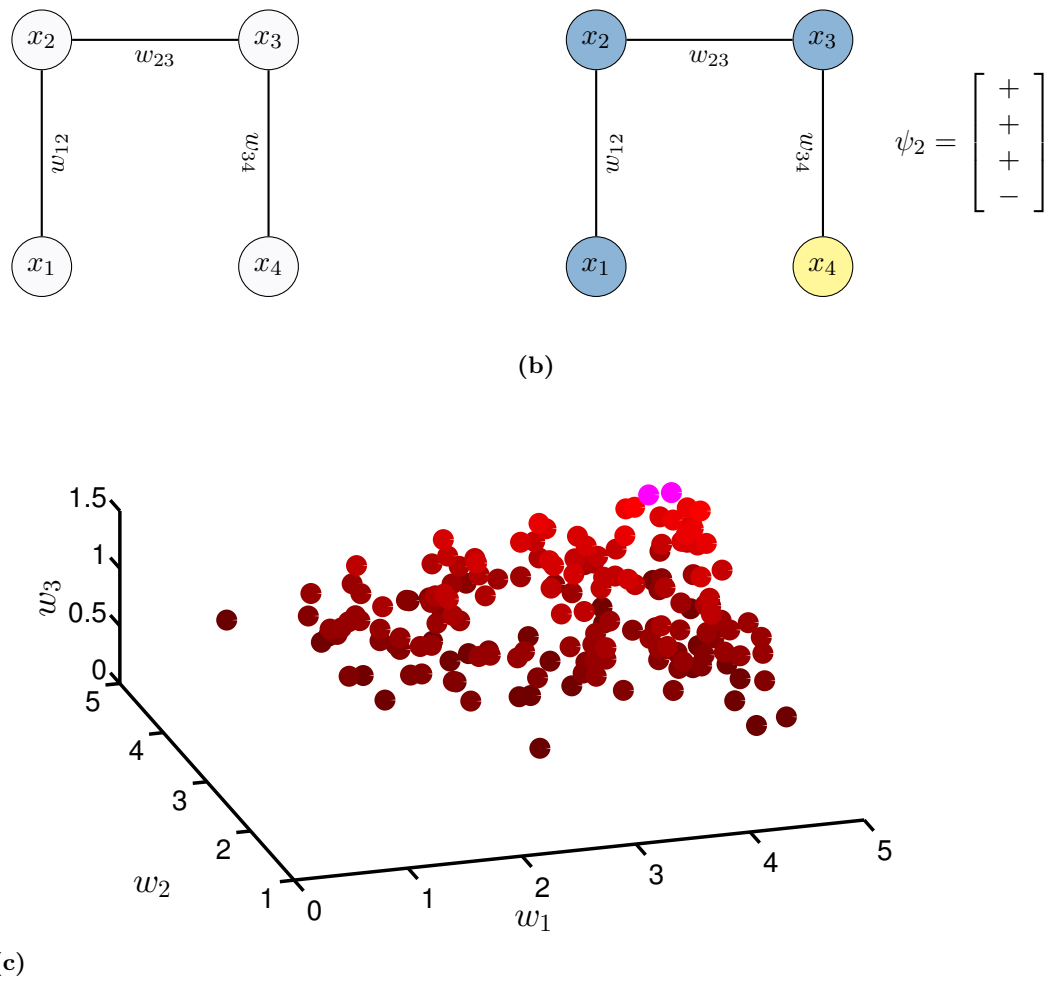


Figure 6.2 (a) Four-node path graph $\mathcal{P}_4$, (b) Nodal domain design choice, (c) Feasible region for the nodal domain constraints.

Table 6.1 Summary of numerical results for $\mathcal{P}_4$.

	$J(\tilde{W}) = \lambda_2$	v_2	$\text{diag}[\tilde{W}]$
Brute force	1.3498	$\begin{bmatrix} 0.429 \\ 0.3083 \\ 0.1052 \\ -0.8425 \end{bmatrix}$	$\begin{bmatrix} 4.8 \\ 4.9 \\ 1.2 \end{bmatrix}$
NLP	1.4488	$\begin{bmatrix} 0.4364 \\ 0.3098 \\ 0.0933 \\ -0.8396 \end{bmatrix}$	$\begin{bmatrix} 4.993 \\ 4.9994 \\ 1.3038 \end{bmatrix}$
Unconstrained	2.9289	$\begin{bmatrix} 0.6533 \\ 0.2706 \\ -0.2706 \\ -0.6533 \end{bmatrix}$	$\begin{bmatrix} 5.0 \\ 5.0 \\ 5.0 \end{bmatrix}$

6.2.2 Four-node Star Graph $\mathcal{S}_4$

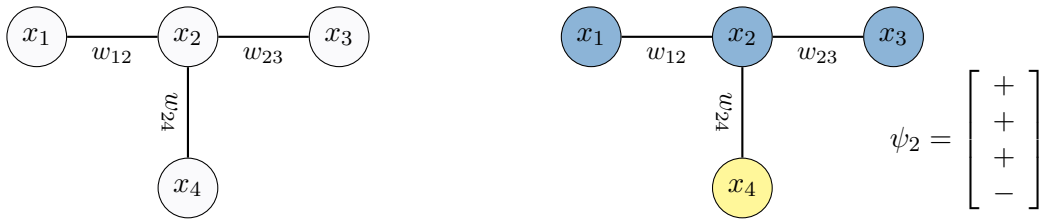
The four-node star graph has seven feasible nodal domain configurations for λ_2, but only two configurations with no zero nodes:

$$\mathcal{C}_2 = \left\{ \begin{bmatrix} + \\ + \\ + \\ - \end{bmatrix}, \begin{bmatrix} + \\ 0 \\ + \\ - \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ + \\ - \end{bmatrix}, \begin{bmatrix} + \\ 0 \\ 0 \\ - \end{bmatrix}, \begin{bmatrix} + \\ 0 \\ - \\ 0 \end{bmatrix}, \begin{bmatrix} + \\ 0 \\ - \\ - \end{bmatrix}, \begin{bmatrix} + \\ - \\ - \\ - \end{bmatrix} \right\}, \tag{6.28a}$$

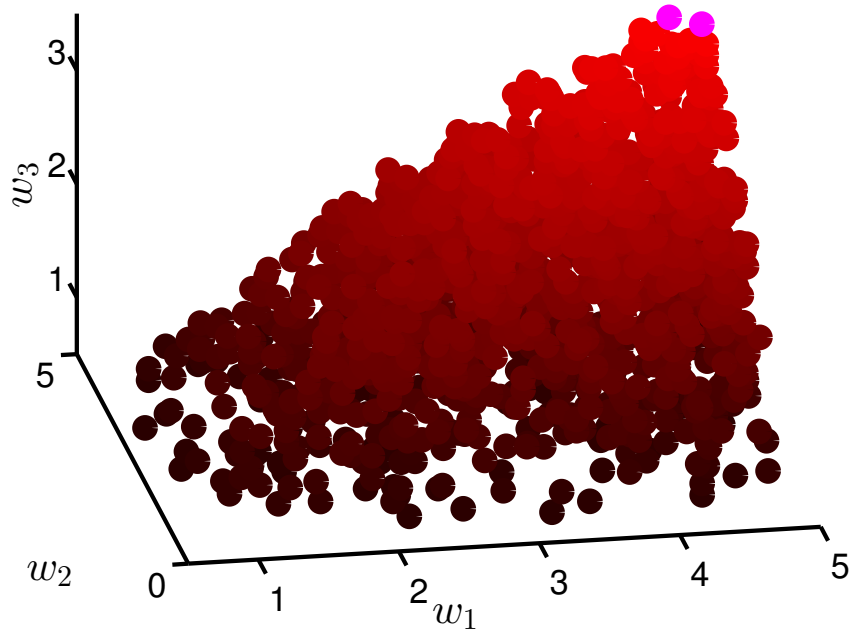
$$\mathcal{C}_2^* = \left\{ \begin{bmatrix} + \\ + \\ + \\ - \end{bmatrix}, \begin{bmatrix} - \\ - \\ + \\ - \end{bmatrix}, \begin{bmatrix} + \\ - \\ - \\ - \end{bmatrix} \right\}. \tag{6.28b}$$

For $\mathcal{S}_4$, we consider the design choice $\psi_2 = [+ \ + \ + \ -]^T$, which is shown in Fig. 6.3(b). We ran experiments identical to those for $\mathcal{P}_4$. Fig. 6.3(c) depicts the feasible region for the $\mathcal{S}_4$ nodal domains of λ_2 given this particular design choice. Table 6.2 summarizes the results for the brute force search and the NLP routine.

In both Tables 6.1 and 6.2 we observe that the unconstrained objective function is greater than the constrained objective. However, in both cases we also observe that resulting eigenvector v_2 of the closed-loop graph does not meet the desired design criteria. Hence, if the desired eigenvectors have been chosen to ensure localizability for security purposes, then the decreased performance must be considered the *cost of security*.



(a) **(b)**



(c)

Figure 6.3 (a) Four-node star graph $\mathcal{S}_4$, (b) Nodal domain design choice, (c) Feasible region for the nodal domain constraints.

Table 6.2 Summary of numerical results for $\mathcal{S}_4$.

	$J(\tilde{W}) = \lambda_2$	v_2	$\text{diag}[\tilde{W}]$
Brute force	3.5635	$\begin{bmatrix} 0.3961 \\ 0.0958 \\ 0.3511 \\ -0.8430 \end{bmatrix}$	$\begin{bmatrix} 4.7 \\ 4.9 \\ 3.2 \end{bmatrix}$
NLP	3.7499	$\begin{bmatrix} 0.3747 \\ 0.0937 \\ 0.3746 \\ -0.8429 \end{bmatrix}$	$\begin{bmatrix} 4.9998 \\ 5.0 \\ 3.3744 \end{bmatrix}$
Unconstrained	5.0	$\begin{bmatrix} 0.3482 \\ 0 \\ -0.8137 \\ 0.4655 \end{bmatrix}$	$\begin{bmatrix} 5.0 \\ 5.0 \\ 5.0 \end{bmatrix}$

6.3 Three-node Path Graph

The three-node path graph is more tractable. Here we show that $\mathcal{W}_\psi$ is convex (in fact, linear) for the three-node path graph. The implication of these results is that we can *efficiently* design an optimal controller K for a *three-area* network where an exogenous disturbance $v(t)$ is guaranteed to be localizable.

Proposition 6.3.1. *Given a three-node graph $\mathcal{P}_3$ and nodal domain configuration $\psi_2 = [+ + -]^T$ (or $\psi_2 = [+ - -]^T$) corresponding to λ_2, then $\tilde{W} \in \mathcal{W}_{\psi_2}$ if and only if $\tilde{w}_{12} > \tilde{w}_{23}$ or ($\tilde{w}_{23} > \tilde{w}_{12}$).*

Proof. This is a special case of Theorem 6.1.1. □

Intuitively recall the results from Theorem 6.1.2, from which we conclude that $\mathcal{P}_3$ has a nodal domain configuration $\psi_2 = [+\,0\,-]^T$ if and only if $\tilde{w}_{12} = \tilde{w}_{23}$. Proposition 6.3.1 states that for the three-area path graph, the nodal domain constraints can be characterized by a linear inequality. In practice, we may add a bias term $0 < \beta < \tilde{a}$ to ensure that $[v_2]_2$ is sufficiently non-zero, in which case the edge weights constraints become

$$\tilde{w}_{12} - \tilde{w}_{23} > \beta \text{ (or } \tilde{w}_{23} - \tilde{w}_{12} > \beta). \tag{6.29}$$

The nodal domain constraints for the three-node path are visualized in Fig. 6.4. We also note the following regarding the nodal domain configurations corresponding to λ_3.

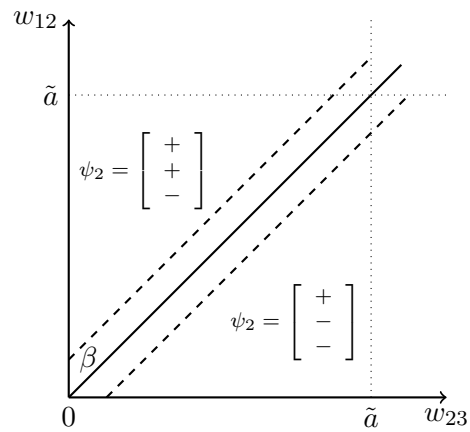


Figure 6.4 Linear inequality constraints that guarantee localizability in the three-node path.

Remark 6.3.1. *In the three-node path $\mathcal{P}_3$, if $\lambda_3 > \lambda_2$, then*

$$\mathcal{C}_3^* = \mathcal{C}_3 = \left\{ \begin{bmatrix} + \\ - \\ + \end{bmatrix} \right\}. \tag{6.30}$$

Proposition 6.3.1 and the previous remark together imply that for the three-node path, $\mathcal{W}_\psi = \mathcal{W}_{\psi_2}$. This is important because it means that we can efficiently guarantee the localizability of disturbances in three-area path networks by designing the aggregate edge weights according to (6.29). This chapter formulated a novel constrained optimal control problem that can ensure localizability of distrubances in the closed-loop graph by shaping its Laplacian eigenvectors. In general such problems are very hard. Utilizing results from graph-theory we were able to find numerical solutions to shaping the λ_2-nodal domains of trees. In the simplest non-trivial tree, $\mathcal{P}_3$, there is an analytical solution. Since the algorithms presented in Part I localize disturbances in the aggregate network, we developed control designs for the reduced order aggregate network. The next chapter illustrates how we can implement these results in the full-order network via wide-area control.

Chapter 7

Composite Design for Control Inversion

The control designs in Chapter 6 were developed for the reduced-order aggregate network. The question, therefore, is how can these designs be implemented in the full-order network so that the control design has the same impact on the slow subsystem's dynamics. In this chapter we develop a framework for practically implementing such aggregate control designs. In order to do this, we recall the standard singular perturbation form of the clustered LTI consensus network (2.25). Our objective is to design $u_s(t)$ in (6.6) and implement it in (6.1a) through $u(t)$ as a state feedback control. We will do this be designing a composite control of the form

$$u(t) = u_s(t) + u_f(t) \tag{7.1}$$

using the slow and fast states $x_s(t)$ and $x_f(t)$ from (6.4) and (6.3), respectively. The motivation for this arises from the fact that typical area-level controllers which employ only local feedback can only effectively influence the fast variables $x_f(t)$. In order to implement wide area control we must somehow incorporate inter-area state information. This requires communication between areas. For consistency, we will assume the network is a 'power system' and, as in Chapter 5, we assume that each area contains a PDC which acts as an area-level coordinator. Note that unlike Chapter 5, we assume that there is no central operator and these coordinators communicate directly with each other. Our approach leverages the singular perturbation form to design a decoupled multi-model control, which further simplifies the problem. The theory and slow and fast subproblems are presented next, followed by a case study which illustrates how the results from this chapter can be combined with the designs from Chapter 6.

7.1 Multi-model Representation

We first review the multi-model representation of our clustered NDS (A.12) [KK78] by exploiting the structure of the singular perturbation form (2.25). Notice that here we assume that node weights are unity. Hence, the singular perturbation parameter in (2.25) becomes $\epsilon_m = \epsilon$ and $d = \overline{d}^E/\underline{d}^I$. The resulting formulation for the kth *simplified model* is simpler that the original model (6.1a) and the singular perturbation model (2.25). This model is an effective representation of the entire network from the perspective of the operator in Area k. To derive the kth simplified model we first rewrite the normalized state matrices in (2.28) explicitly as

$$\overline{A}_{12} = [A_{o1}, \ldots, A_{op}], \qquad A_{oi} \in \mathbb{R}^{p\times(n_i-1)}, \tag{7.2a}$$

$$\overline{A}_{21}^T = [A_{1o}, \ldots, A_{po}], \qquad A_{io} \in \mathbb{R}^{(n_i-1)\times p}, \tag{7.2b}$$

$$\overline{A}_{22} = A_{22o} + dA_{22d}, \tag{7.2c}$$

$$\overline{B}_1 = \mathrm{diag}[B_s^1, \ldots, B_s^p], \qquad B_i \in \mathbb{R}^{1\times n_i}, \tag{7.2d}$$

$$\overline{B}_2 = \mathrm{diag}[B_f^1, \ldots, B_f^p], \qquad B_f^i \in \mathbb{R}^{(n_i-1)\times n_i}, \tag{7.2e}$$

where

$$A_{22o} = \mathrm{diag}[A_{22o}^1, \ldots, A_{22o}^p] \tag{7.3}$$

$$A_{22d} = \mathrm{diag}[A_{22d}^1, \ldots, A_{22d}^p] + \begin{bmatrix} 0_{(n_1-1)\times(n_1-1)} & \tilde{A}_{12} & \cdots & \tilde{A}_{1p} \\ \tilde{A}_{21} & 0_{(n_2-1)\times(n_2-1)} & & \tilde{A}_{2p} \\ \vdots & & \ddots & \vdots \\ \tilde{A}_{p1} & \cdots & \tilde{A}_{p(p-1)} & 0_{(n_p-1)\times(n_p-1)} \end{bmatrix}. \tag{7.4}$$

Using the representation in (7.2), we can rewrite the singular perturbation form (2.25) as

$$\hat{M}\dot{x}_s(t) = \overline{A}_{11}x_s(t) + \sum_{k=1}^{p} A_{ok}x_f^k(t) + \sum_{k=1}^{p} B_{ok}u^k(t) \tag{7.5a}$$

$$\epsilon\dot{x}_f^k(t) = dA_{ko}x_s(t) + A_f^k x_f^k(t) + d\sum_{j\neq k}^{p} \tilde{A}_{kj}x_f^j(t) + B_f^k u^k(t), \tag{7.5b}$$

for $k = 1, \ldots, p$, where $x_f^k \in \mathbb{R}^{n_k-1}$ denotes the fast states of the kth area, $u^k \in \mathbb{R}^{n_k}$ denotes the control inputs in the kth area, $A_f^k \triangleq A_{22o}^k + dA_{22d}^k$, and $B_{ok} \triangleq \mathbf{e}_k \otimes B_s^k \in \mathbb{R}^{p\times n_k}$.

We next assume that the decision maker in the kth area neglects the weak connections between other areas as well as the fast dynamics of all other subsystems. Mathematically this is

equivalent to setting $dA_{ko} = d\tilde{A}_{ki} = 0$ in (7.5a) and $\epsilon x_f^i = 0,\ \forall i \neq k$ in (7.5b). The resulting model becomes

$$\hat{M}\dot{x}_s(t) = \overline{A}_{11}x_s(t) + \sum_{k=1}^{p} A_{ok}x_f^k(t) + \sum_{k=1}^{p} B_{ok}u^k(t) \tag{7.6a}$$

$$\epsilon\dot{x}_f^k(t) = A_f^k x_f^k(t) B_f^k u^k(t), \tag{7.6b}$$

$$0 = A_f^i x_f^i(t) + B_f^i u^i(t), \qquad i = 1, \ldots, p,\ i \neq k. \tag{7.6c}$$

If A_f^i is nonsingular, we can substitute $x_f^i(t) = -(A_f^i)^{-1}B_f^i u^i(t)$ into (7.6a) to obtain the kth *simplified model* as

$$\hat{M}\dot{x}_s^k(t) = \overline{A}_{11}x_s^k(t) + A_{ok}x_f^k(t) + B_{ok}u^k(t) + \sum_{j\neq k}^{p} \tilde{B}_s^j u^j(t) \tag{7.7a}$$

$$\epsilon\dot{x}_f^k(t) = A_f^k x_f^k(t) + B_f^k u^k(t), \tag{7.7b}$$

where $\tilde{B}_s^j \triangleq B_{oj} - A_o j(A_f^j)^{-1}B_f^j$. The superscript k indicates that the states in (7.7) are perceived by the decision maker in the kth area. Finally, it is worth mentioning that the use of A_f^k, opposed to simply using A_{22o}^k, will enhance the performance of the model, especially when d is not sufficiently small [CK85]. We next use this multi-model approach to design p unique controllers, one for each of the p aggregate areas.

7.2 Multi-area Control

We are now ready to design the composite control $u^k(t) \in \mathbb{R}^{n_k}$ in (7.1) for each area, which assumes the form

$$u^k(t) = u_s^k(t) + u_f^k(t), \quad k = 1, \ldots, p, \tag{7.8}$$

where $u_s^k(t) \in \mathbb{R}^{n_k}$ and $u_f^k(t) \in \mathbb{R}^{n_k}$ represent the vectors of slow control input and fast control input, respectively, for each node in the kth area. Recall that the purpose of u_s^k is to influence the slow time-scale dynamics of the aggregate network (6.4), while u_f^k is a high-gain control input designed to drive $\dot{x}_f^k(t) \to 0$ quickly.

Assumption 7.2.1. *Each agent in (6.1a) has a single control input. Without loss of generality, we can set $B = I_n$.*

7.3 Fast Control Design

In order to design the fast control $u_f^k(t)$ we consider the fast time-scale $t_f = t/\epsilon$ and let $\epsilon \to 0$ in (7.7). The resulting fast state dynamics for the kth area become

$$\frac{d}{dt_f} x_f^k(t_f) = A_f^k x_f^k(t_f) + B_f^k u^k(t_f) = A_f^k x_f^k(t_f) + B_f^k(u_f^k(t_f) + u_s^k(t_f)). \tag{7.9}$$

In order to decouple the fast and slow control designs we must design u_s^k such that it is in the nullspace of B_f^k, i.e. such that $B_f^k u_s^k = 0_{n_k}$. In light of Assumption 7.2.1, the form of B_f^k in (7.2e) becomes $B_f^k = F^k$ defined in (2.17), whose nullspace is spanned by $\mathbf{1}_{n_k}$. Hence, the slow control input should take the form

$$u_s^k(t) = \mathbf{1}_{n_k} \hat{u}_s^k(t), \tag{7.10}$$

for $k = 1, \ldots, p$, where $\hat{u}_s^k(t) \in \mathbb{R}$. In addition to decoupling the fast and slow controls, (7.10) clearly further simplifies the slow control design. We will discuss the implications of this in a moment, after we complete the fast control design. Substituting (7.10) into (7.9) yields

$$\frac{d}{dt_f} x_f^k(t_f) = A_f^k x_f^k(t_f) + B_f^k u_f^k(t_f). \tag{7.11}$$

Finally, writing

$$u_f^k(t_f) = K_f^k x_f^k(t_f), \tag{7.12}$$

where $K_f^k \in \mathbb{R}^{n_k \times (n_k - 1)}$ is the fast control gain matrix, results in

$$\frac{d}{dt_f} x_f^k(t_f) = (A_f^k + B_f^k K_f^k) x_f^k(t_f). \tag{7.13}$$

We can now state the fast control design problem precisely as:

Given a lower bound $\underline{\lambda} \in \mathbb{R}$, find the gains K_f^k such that the eigenvalues of $(A_f^k + B_f^k K_f^k)$ satisfy

$$\underline{\lambda} < \lambda_i(A_f^k + B_f^k K_f^k), \; i = 1, \ldots, n_k - 1, \tag{7.14}$$

for $k = 1, \ldots, p$.

The problem (7.14) is a standard pole-placement problem. The parameter $\underline{\lambda}$ controls the width of the gap between the fast and slow eigenvalues of the closed loop system (A.12). Hence we require that $\underline{\lambda} > \lambda_p^*$, where λ_p^* is the fastest eigenvalue resulting from the slow control design,

which we describe next.

7.3.1 Slow Control Design

For the slow control design, recall that we require $u_s^k = \mathbf{1}_{n_k}\hat{u}_s^k$ for $k = 1, \ldots, p$. This simplifies the design because we now have a single slow control input to design for each area, which then gets mapped to every agent within the area. While it also initially seems to restrict our design, notice that in the slow time-scale the inputs get aggregated. Thus there is no advantage to designing n_k slow control inputs for the kth area. Concretely, to design the slow control we let $\epsilon \to 0$ in the kth simplified model (7.7), which results in

$$\hat{M}\dot{x}_s(t) = \overline{A}_{11}x_s(t) + \sum_{j=1}^{p} \tilde{B}_s^j u_s^j \tag{7.15a}$$

$$x_f^k(t) = -(A_f^k)^{-1}B_f^k u_s^k(t). \tag{7.15b}$$

Notice that since we let $\epsilon \to 0$ for all Areas $k = 1, \ldots, p$, the aggregate model (7.15a) is no longer exclusive to the kth area, and thus we can drop the superscript on $x_s(t)$. Next, we substitute $u_s^k = \mathbf{1}_{n_k}\hat{u}_s^k$, and recall that $\tilde{B}_s^j = B_{oj} - A_{oj}(A_f^j)^{-1}B_f^j$ and $B_f^j\mathbf{1}_{n_j} = 0_{n_j}$, so (7.15) becomes

$$\hat{M}\dot{x}_s(t) = \overline{A}_{11}x_s(t) + \sum_{k=1}^{p} B_{ok}\mathbf{1}_{n_k}\hat{u}_s^k. \tag{7.16}$$

Furthermore, since $B_{ok} = \mathbf{e}_k \otimes B_s^k$, (7.16) reduces to

$$\hat{M}\dot{x}_s(t) = \overline{A}_{11}x_s(t) + \hat{B}_s\hat{u}_s, \tag{7.17}$$

where $\hat{B}_s = \mathrm{diag}[\hat{b}_1, \ldots, \hat{b}_p] \in \mathbb{R}^{p\times p}$. Using (2.28) and Assumption 7.2.1 we have that

$$\hat{b}^k = \frac{\mathbf{1}_{n_k}^T I_{n_k}\mathbf{1}_{n_k}}{\overline{\gamma}^E} = \frac{n_k}{\overline{\gamma}^E}, \tag{7.18}$$

where $\overline{\gamma}^E$ is defined in (2.27). To simplify notation, we define the effective slow control input as

$$\bar{u}_s(t) \triangleq \hat{B}_s\hat{u}_s(t), \tag{7.19}$$

where $\overline{u}_s(t) = [\overline{u}_s^1(t), \ldots, \overline{u}_s^p(t)]^T \in \mathbb{R}^p$. Hence, (7.17) becomes

$$\hat{M}\dot{x}_s(t) = \overline{A}_{11}x_s(t) + \bar{u}_s(t). \tag{7.20}$$

Recall that $\overline{A}_{11}$ is precisely the graph Laplacian matrix of the aggregate graph $\mathcal{G}$. Hence, (7.20) is precisely in the form of (6.4) and we can design $\bar{u}_s$ as in Chapter 6. In general, we write $\bar{u}_s(t)$ as a function of the slow states $x_s(t)$ as

$$\bar{u}_s(t) = \overline{K}_s x_s(t), \tag{7.21}$$

where $\overline{K}_s \in \mathbb{R}^{p\times p}$ is the gain matrix for the slow control design. For example, in Chapter 6, $\overline{K}_s$ was a control graph Laplacian defined as $\overline{K}_s = -L_{\mathcal{K}}$. In order to implement the slow control design $\overline{K}_s$ in the network, we must map it back to $u_s(t)$. Hence the term *control inversion*. This is accomplished by stacking the slow inputs $u_s^k(t)$ in (7.10) for every area, which yields

$$u_s(t) = \begin{bmatrix} u_s^1(t) \\ \vdots \\ u_s^p(t) \end{bmatrix} = U\hat{u}_s(t), \tag{7.22}$$

where $U = \text{diag}[\mathbf{1}_{n_1}, \ldots, \mathbf{1}_{n_p}]$. Since Assumption 7.2.1 implies $\hat{B}$ is invertible, we use (7.19) to find

$$\hat{u}_s(t) = \hat{B}^{-1}\bar{u}_s(t). \tag{7.23}$$

Combining (7.23) with (7.21) in (7.22) results in implementable form

$$u_s(t) = \underbrace{U\hat{B}^{-1}\overline{K}_s}_{K_s} x_s(t). \tag{7.24}$$

It is important to note that $\overline{K}_s$ (and hence K_s) must be designed cooperatively by the coordinators in each area. However, when implementing the slow control each coordinator only requires his block $u_s^k(t)$, which can be succinctly written as

$$u_s^k(t) = K_s^k x_s(t), \tag{7.25}$$

where $K_s^k \in \mathbb{R}^{n_k\times p}$ is defined as

$$K_s^k \triangleq [\hat{B}^{-1}\overline{K}_s]_k \otimes \mathbf{1}_{n_k}, \tag{7.26}$$

with $[\hat{B}^{-1}\overline{K}_s]_k \in \mathbb{R}^{1\times p}$ denoting the kth row of the matrix $\hat{B}^{-1}\overline{K}_s$. We next show how to combine the slow and fast control designs and provide performance bounds of our composite control design.

7.3.2 Performance of the Multi-Model Control

Using the results from the fast and slow control designs, the composite control (7.1) can now be written as

$$u(t) = K_s x_s(t) + K_f x_f(t), \tag{7.27}$$

where $K_s \in \mathbb{R}^{n\times p}$ is defined in (7.24) and $K_f = \text{diag}[K_f^1, \ldots, K_f^p] \in \mathbb{R}^{n\times(n-p)}$ is specified in (7.12). Since $u^k(t)$ is designed from the perspective Area k without regard to the other areas, we refer to (7.27) as the *decoupled design*. The following theorem bounds the performance error of using our decoupled multi-model approach to design $u^k(t),\ k = 1, \ldots, p$.

Remark 7.3.1. *Assumption 7.2.1 implies that the aggregate network is completely controllable, i.e.* $(\overline{A}_{11}, I_p)$ *is controllable.*

Theorem 7.3.1. *Let* $\zeta = [\epsilon\ d]^T \in \Omega \subset \mathbb{R}^2$. *If the fast pairs* (A_f^k, B_f^k) *for* $k = 1, \ldots, p$, *are controllable and* $K_s,\ K_f^k$ *are designed to assign distinct, stable eigenvalues* $\lambda_i(\overline{A}_{11} + \overline{K}_s),\ i = 1, \ldots, p$, *and* $\lambda_i(A_f^k + B_f^k K_f^k),\ i = 1, \ldots, n_k - 1$, *respectively, then there exists* $\alpha > 0$ *such that* $\forall \zeta \in \Omega$ *with* $0 < \|\zeta\| \le \alpha$, *the closed-loop system (6.1a) has* p *slow eigenvalues* $\{\lambda_1^c, \ldots, \lambda_p^c\}$ *and* $(n-p)$ *fast eigenvalues* $\{\lambda_{p+1}^c, \ldots, \lambda_n^c\}$ *which are approximated by*

$$\lambda_i^c = \lambda_i(\overline{A}_{11} + \sum_{j=1}^{p} \tilde{B}_j^k K_j^k) + \mathcal{O}(\|\zeta\|),\ i = 1, \ldots, p \tag{7.28}$$

$$\lambda_i^c = \left[\lambda_j(A_f^1 + B_f^1 K_f^1) + \mathcal{O}(\|\zeta\|)\right]/\epsilon,\ i = p + j,\ j = 1, \ldots, n_1 - 1 \tag{7.29}$$

$$\vdots$$

$$\lambda_i^c = \left[\lambda_l(A_f^p + B_f^p K_f^p) + \mathcal{O}(\|\zeta\|)\right]/\epsilon,\ i = p + n_{p-1} + l - 1,\ l = 1, \ldots, n_p - 1. \tag{7.30}$$

Proof. See Appendix A.3. □

7.4 Case Study

This section illustrates the multi-model control design for a clustered network. We demonstrate the general performance as described by Theorem 7.3.1, and show how the design for $\mathcal{P}_3$ from Chapter 6 can be implemented in a three-area NDS using wide-area control inversion.

7.4.1 Three-Area Case Study

We consider the three-area 30-node network shown in Figure 7.1, with consensus dynamics described by (6.1a), and design a composite control $u(t) = u_s(t) + u_f(t)$ which solves problem

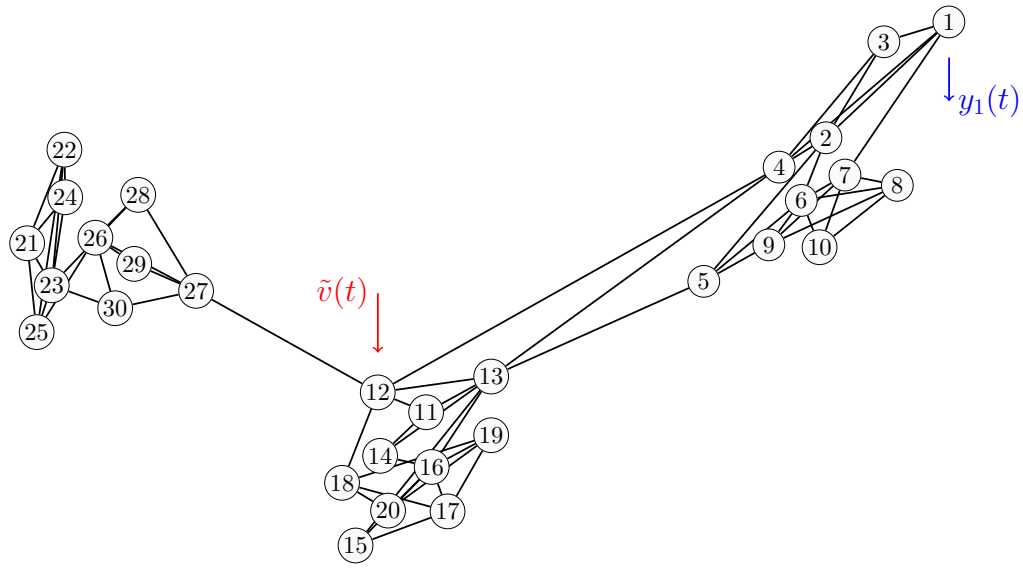


Figure 7.1 Three-area 30-node network used in the simulation; the areas are designated as $\mathcal{V}^1 = \{1, \ldots, 10\}$, $\mathcal{V}^2 = \{11, \ldots, 20\}$, $\mathcal{V}^3 = \{21, \ldots, 30\}$.

(6.26). Recall that the objectives of (6.26) were twofold. First, we would like to ensure localizability of disturbances in the aggregate subsystem. Second, we would like to maximize the overall consensus speed. We compare the results of naive structured optimal control design that simply attempts to maximize the consensus speed and our proposed approach (6.26) with varying levels of bias β.

The first step of our control design is to aggregate the network in Fig. 7.1. Area 1 consists of nodes $\mathcal{V}^1 = \{1, \ldots, 10\}$, Area 2 is $\mathcal{V}^2 = \{11, \ldots, 20\}$, and likewise $\mathcal{V}^3 = \{21, \ldots, 30\}$. This results in a three-node path graph as shown in Fig. 6.1(a) with edge weights

$$W = \text{diag}[0.20,\ 0.10]. \tag{7.31}$$

The nominal open-loop eigenvalues of the slow subsystem are

$$\lambda_{ol}^2 = \{0,\ 0.08,\ 0.25\}. \tag{7.32}$$

We set the upper bound on the control graph edge weights to be $\tilde{a} = 2.0$ per-unit. We find the optimal slow state feedback controller without the nodal domain constraints by solving (6.26) but letting $\beta = 0$. This results in an optimal control gains of

$$K^* = \text{diag}[1.80,\ 1.90]. \tag{7.33}$$

The sign pattern of the corresponding closed-loop eigenvectors is

$$\operatorname{sgn}(v_2) = \begin{bmatrix} + \\ 0 \\ - \end{bmatrix}. \tag{7.34}$$

Theoretically, this means that the transfer function corresponding to the slow subsystem will not be minimal, according to Theorem 3.2.1. Hence an operator would not be able to localize a disturbance that enters Area 2 using the algorithm in [NC13].

The fast state feedback gains $K_f^k \in \mathbb{R}^{10\times 9}$ are found using the Matlab function `place`. The gains are chosen such that the fast eigenvalues are on the order of ten times faster than the fastest closed-loop slow eigenvalue. The slow closed-loop eigenvalues of this design are

$$\lambda_{cl}^s = \{0,\ 1.84,\ 5.53\}. \tag{7.35}$$

The elements of the eigenvector corresponding λ_2 of this design are plotted in Fig. 7.2. As expected from the aggregate graph design, the elements corresponding to Area 2 are very close to zero when $\beta = 0$. Indeed, if we assume an impulsive disturbance occurs at node $12 \in \mathcal{V}^2$ and consider an output from node $1 \in \mathcal{V}^1$, we find the transfer function (without considering measurement noise) to be

$$g_{1,12}(s) = \frac{0.033}{s} + \frac{-0.0002}{s+1.84} + \frac{-0.032}{s+5.53} + g^{fast}(s), \tag{7.36}$$

where the fast component has been truncated in $g^{fast}(s)$. The contribution of the mode $\lambda_2 = 1.84$ to the transfer function is practically negligible. As a result, if a network operator is using measurements $y_1(t)$, it becomes impossible to accurately localize the disturbance.

We find the optimal slow state feedback controller with the nodal domain constraints by solving (6.26). We choose $\psi_2 = [+, +, -]^T$ and use the constraints (6.29). Since the problem is still convex, we can use convex optimization packages such as `cvx` to solve the problem. We repeat the design procedure for β varying from 0.5 to 1.5. As β increases, the elements of v_2 corresponding to Area 2 move farther from zero. However, the closed loop convergence rate, which is bounded by λ_2, is adversely affected. Indeed, this is the price of localizability. Table 7.1 summarizes the performance degradation as β is increased. Note that so long as the elements of v_2 are sufficiently non-zero, depending on the amount of noise present, the disturbances will be localizable. Letting $\beta = 0.5$ may be sufficient, in which case the price of ensuring localizability of the closed-loop system is rather small (compare $\lambda_2 = 1.84$ to $\lambda_2 = 1.56$).

Table 7.1 The price of localization: as β increases, consensus speed decreases.

β	λ_2	λ_3
0	1.84	5.53
0.5	1.56	4.81
1.0	1.13	4.23
1.5	0.56	3.81

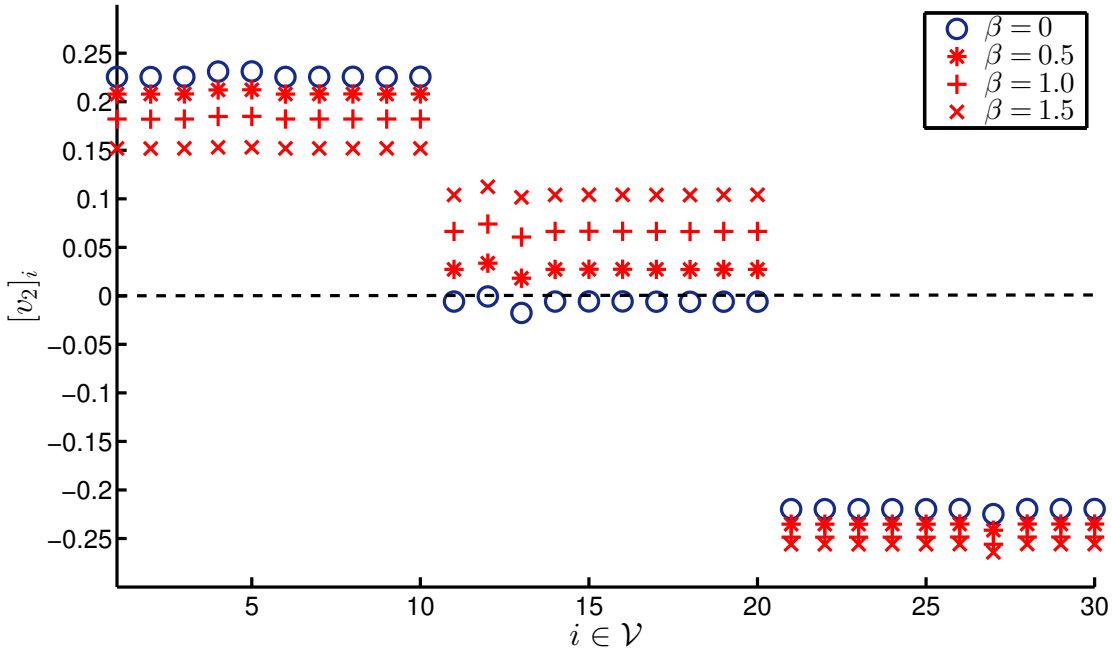


Figure 7.2 Values of v_2 of the optimal closed loop graph with varying levels of the bias term β.

Chapter 8

Conclusions

Measurement-based, data-driven algorithms are rapidly becoming more popular for network operators, driven by the growth of network size and the explosion of sensor data. Simultaneously, with the need to analyze gigantic amounts of data in real-time, operators are persistently searching for efficient numerical algorithms by which they can monitor the health of their network following critical disturbances or faults. In this thesis, we developed a suite of algorithms for one such specific health monitoring application: namely, localization of disturbances in large graphs, and ensuring localizability via closed-loop state-feedback-based optimal control. These algorithms find particular relevance in the monitoring and control of large-scale power systems equipped with WAMS, where security on both the physical and cyber level is becoming increasingly important.

In Part I we developed a measurement-based graph-theoretic algorithm to localize disturbances in a large NDS. In doing so, we drew new connections between the properties of the underlying graph and the measurable dynamics. Namely, the connection between the sign pattern of the system's transfer function residues and the relative input-output locations with respect to the graph-theoretic nodal domains. The algorithm itself was built on a clustering/aggregation assumptions of the large-scale network, and hence very efficient. We developed theoretical extensions of our basic algorithm and implemented a real-time version in a realistic power system example.

In Part II we formulated new constraints for optimal control problems, whereby the network operator can guarantee localizability of disturbances in the closed-loop system. Using these constraints, we developed optimal control designs for aggregate networks that are trees. In order to implement these control designs, we drew inspiration from classical multi-model control design to develop a control inversion technique in the completely new context of wide-area control.

8.1 Future Directions

This thesis can in no-way completely tackle the problem of monitoring and control of networked dynamic systems. The introduction of nodal domains into the equation—particularly the control designs—may have introduced even more questions than we have addressed. The following includes some high-level descriptions of these open problems and future directions for this research.

8.1.1 Monitoring

Sensitivity Analysis of Localization

In Chapter 4 we demonstrated the effectiveness of our input localization under a certain level of measurement noise. However, no rigorous study was undertaken to show how sensitive the localization algorithm is to process and measurement noise. Understanding how sensitive the localization algorithm is on a whole, and how sensitive are its individual parts—system identification, localization keys, and residues—will be critical to effectively deploying such an algorithm in a real-world system. Furthermore, it may be possible to infer information about the disturbance from a set of *partial* or *fuzzy* localization keys in particularly noisy or poorly clustered networks. That is, using the techniques developed in this thesis alongside tools for statistical machine learning.

Distribute Localization

In Chapter 5 we developed a real-time version of our localization algorithm that relied on a centralized independent system operator. This formulation was attractive because of its simplicity and the privacy preserving nature of such an architecture. However, as with most centralized algorithms, it is not a stretch to imagine that our real-time localization algorithm can be cast as a distributed consensus problem. In fact, the first step in the localization algorithm, system identification, has recently been distributed [NC14].

Granular Localization

One of the major limitations of our localization algorithm is the lack of granularity, namely, it can only localize a disturbance up to the cluster level. In order to achieve a more granular localization we could incorporate information from the fast modes along with other information that can be obtained or inferred from the measured outputs. An example includes distance-based information, such as electrical distances which are also closely linked to transfer function residues. Such an approach would undoubtedly require the help of statistical learning.

An alternative approach would be to try to utilize more powerful mathematical tools from Homology, which can utilize higher-powered Laplacians, and Sheaf theory [Rob13; Cur13].

8.1.2 Control

Nodal Domain Characterization

One of the first major questions that needs to be addressed is how to systematically find implementable constraints for nodal domain configurations other than ψ_2, and for graphs other than just trees. Characterization of Laplacian eigenvectors other than v_2 seems to be a largely open question in the graph theory community, although new results and algorithms are developed constantly.

Game Theoretic Considerations

The proposed control problem ensures localizability of an attack by augmenting the underlying network via differential state feedback. But as we noticed, there may be multiple closed loop graphs to choose from that all ensure localizability. With the increasing sophistication of cyber-physical attacks, it may be advantageous for the network operator to alternate between feasible nodal domain configurations to fool the attacker. The attack scenario may, therefore, be posed as a zero-sum game in the form of a pursuit-evasion problem.

Control Inversion in Higher-Order Power Systems

In Chapter 7 we developed a control inversion technique for implementing aggregate control designs. One of the primary motivators for this work is its application in large power system networks. Our technique leveraged the first-order consensus NDS. In this case, the measured outputs and control inputs are directly related to dynamics that we wish to shape. However, in a realistic power system network, the control input enters through a synchronous generator's exciter voltage, while the control objective is to shape the inter-area modes, or frequencies. Hence, the control inversion presented here must be rigorously extended to *at least* a third-order model of the power system network.

BIBLIOGRAPHY

[AF03] Anderson, P. M. & Fouad, A. A. *Power System Control and Stability*. Second. Piscataway: Wiley-IEEE Press, 2003.

[AM05] Antsaklis, P. J. & Michel, A. N. *Linear Systems*. Boston: Birkhäuser, 2005.

[Ath79] Athay, T. et al. "A Practical Method for the Direct Analysis of Transient Stability". *Power Apparatus and Systems, IEEE Transactions on* **98**.2 (1979), pp. 573–584.

[BA98] Balch, T. & Arkin, R. "Behavior-based Formation Control for Multirobot Teams". *Robotics and Automation, IEEE Transactions on* **14**.6 (1998), pp. 926–939.

[Bal04] Ballance, J. et al. "Use of Synchronized Phasor Measurement System for Enhancing AC-DC Power System Transmission Reliability and Capability". *Eastern Interconnection Phasor Project (EIPP) Meeting*. 2004.

[Ban06] Band, R. et al. "Nodal domains on isospectral quantum graphs: the resolution of isospectrality?" *Journal of Physics A: Mathematical and General* **39**.45 (2006), pp. 13999–14014.

[Biy07] Biyikoğlu, T. et al. *Laplacian Eigenvectors of Graphs*. Ed. by Morel, J. et al. Springer, 2007.

[Bol98] Bollobás, B. *Modern Graph Theory*. New York: Springer, 1998.

[Cha13] Chakrabortty, A. et al. *Wide-Area Monitoring and Control of WECC Transfer Paths Using Real-Time Digital Simulations*. Tech. rep. FREEDM Systems Center at North Carolina State University, 2013.

[CM13] Chapman, A. & Mesbahi, M. "Semi-Autonomous Consensus: Network Measures and Adaptive Trees". *Automatic Control, IEEE Transactions on* **58**.1 (2013), pp. 19–31.

[CK85] Chow, J. & Kokotovic, P. "Time Scale Modeling of Sparse Dynamic Networks". *Automatic Control, IEEE Transactions on* **30**.8 (1985), pp. 714–722.

[Cho95] Chow, J. et al., eds. *Systems and Control Theory for Power Systems*. New York: Springer-Verlag, 1995.

[Cur13] Curry, J. "Sheaves, cosheaves and applications". *arXiv preprint arXiv:1303.3255* (2013).

[DD03] D'Andrea, R. & Dullerud, G. "Distributed control design for spatially interconnected systems". *Automatic Control, IEEE Transactions on* **48**.9 (2003), pp. 1478–1495.

[Des01] Desai, J. P. et al. "Modeling and control of formations of nonholonomic mobile robots". *Robotics and Automation, IEEE Transactions on* **17**.6 (2001), pp. 905–908.

[DB13] Dorfler, F. & Bullo, F. "Kron Reduction of Graphs With Applications to Electrical Networks". *Circuits and Systems I: Regular Papers, IEEE Transactions on* **60**.1 (2013), pp. 150–163.

[Far14] Fardad, M. et al. "Design of Optimal Sparse Interconnection Graphs for Synchronization of Oscillator Networks". *IEEE Transactions on Automatic Control* (2014), pp. 1–1.

[FR11] Farwell, J. P. & Rohozinski, R. "Stuxnet and the Future of Cyber War". *Survival* **53**.1 (2011), pp. 23–40.

[FM04] Fax, J. & Murray, R. "Information Flow and Cooperative Control of Vehicle Formations". *Automatic Control, IEEE Transactions on* **49**.9 (2004), pp. 1465–1476.

[GN02] Girvan, M. & Newman, M. E. J. "Community structure in social and biological networks". *Proceedings of the National Academy of Sciences* **99**.12 (2002), pp. 7821–7826.

[Gor92] Gordon, C. et al. "One cannot hear the shape of a drum". *Bulletin of the American Mathematical Society* **27**.1 (1992), pp. 134–138.

[Hau90] Hauer, J. et al. "Initial results in Prony analysis of power system response signals". *Power Systems, IEEE Transactions on* **5**.1 (1990), pp. 80–89.

[HJ85] Horn, R. A. & Johnson, C. R. *Matrix Analysis*. New York: Cambridge University Press, 1985.

[HJ91] Horn, R. A. & Johnson, C. R. *Topics in Matrix Analysis*. New York: Cambridge University Press, 1991.

[Jad03] Jadbabaie, A. et al. "Coordination of groups of mobile autonomous agents using nearest neighbor rules". *Automatic Control, IEEE Transactions on* **48**.6 (2003), pp. 988–1001.

[JP85] Juang, J.-N. & Pappa, R. "An Eigensystem Realization Algorithm for Modal Parameter Identification and Model Reduction". *Journal of Guidance, Control, and Dynamics* **8**.5 (1985), pp. 620–627.

[Kac66] Kac, M. "Can one hear the shape of a drum?" *The American Mathematical Monthly* **73**.4 (1966), pp. 1–23.

[KK78] Khalil, H. K. & Kokotovic, P. "Control strategies for decision makers using different models of the same system". *Automatic Control, IEEE Transactions on* **23**.2 (1978), pp. 289–298.

[Kir96] Kirkland, S. et al. "Characteristic vertices of weighted trees via perron values". *Linear and Multilinear Algebra* **40**.4 (1996), pp. 311–325.

[Kok99] Kokotovic, P. et al. *Singular Perturbation Methods in Control: Analysis and Design*. London: SIAM, 1999.

[KS72] Kwakernaak, H & Sivan, R. *Linear Optimal Control Systems*. New York: Wiley-Interscience, 1972.

[Lee08] Lee, E. "Cyber Physical Systems: Design Challenges". *Object Oriented Real-Time Distributed Computing, 2008 11th IEEE International Symposium on*. 2008, pp. 363–369.

[Lin11] Lin, F. et al. "Augmented Lagrangian approach to design of structured optimal state feedback gains". *Automatic Control, IEEE Transactions on* **56**.12 (2011), pp. 2923–2929.

[Mac14] Mackenzie, D. "Kadison-Singer Problem Solved". *SIAM News* **47**.1 (2014).

[ML89] Martins, N. & Lima, L. "Determination of Suitable Locations for Power System Stabilizers and Static VAr Compensators for Damping Electromechanical Oscillations in Large Scale Power Systems". *Power Industry Computer Application Conference*. 1989, pp. 74–82.

[ME10] Mesbahi, M. & Egerstedt, M. *Graph Theoretic Methods in Multiagent Networks*. Princeton: Princeton University Press, 2010.

[NC14] Nabavi, S. & Chakrabortty, A. "A real-time distributed Prony-based algorithm for modal estimation of power system oscillations". *American Control Conference (ACC), 2014*. 2014, pp. 729–734.

[NC13] Nudell, T. R. & Chakrabortty, A. "A Graph-Theoretic Algorithm for Disturbance Localization in Large Power Grids Using Residue Estimation". *The 2013 American Control Conference (ACC), Proceedings of*. 2013.

[OS06] Olfati-Saber, R. "Flocking for Multi-agent Dynamic Systems: Algorithms and Theory". *Automatic Control, IEEE Transactions on* **51**.3 (2006), pp. 401–420.

[OS07] Olfati-Saber, R. et al. "Consensus and Cooperation in Networked Multi-Agent Systems". *Proceedings of the IEEE* **95**.1 (2007), pp. 215–233.

[Pas13] Pasqualetti, F. et al. "Attack Detection and Identification in Cyber-Physical Systems". *Automatic Control, IEEE Transactions on* **58**.11 (2013), pp. 2715–2729.

[Pha83] Phadke, A. et al. "A New Measurement Technique for Tracking Voltage Phasors, Local System Frequency, and Rate of Change of Frequency". *Power Apparatus and Systems, IEEE Transactions on* **102**.5 (1983), pp. 1025–1038.

[Rob13] Robinson, M. "Understanding networks and their behaviors using sheaf theory". *arXiv preprint arXiv:1308.4621* (2013).

[Sch12] Schuler, S. et al. "Design of sparse relative sensing networks". *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. 2012, pp. 2749–2754.

[Sha12a] Shafi, S. Y. et al. "Graph Weight Allocation to Meet Laplacian Spectral Constraints". *Automatic Control, IEEE Transactions on* **57**.7 (2012), pp. 1872–1877.

[Sha12b] Shames, I. et al. "Distributed Fault Detection and Isolation with imprecise network models". *The 2012 American Control Conference (ACC)*. 2012, pp. 5906–5911.

[SS11] Spielman, D. A. & Srivastava, N. "Graph Sparsification by Effective Resistances". *SIAM Journal on Computing* **40**.6 (2011), pp. 1913–1926.

[TM01] Tisseur, F. & Meerbergen, K. "The Quadratic Eigenvalue Problem". *SIAM Review* **43**.2 (2001), pp. 235–286.

[TB86] Tsitsiklis, J. & Bertsekas, D. "Distributed asynchronous optimal routing in data networks". *Automatic Control, IEEE Transactions on* **31**.4 (1986), pp. 325–332.

[VODM96] Van Overschee, P. & De Moor, B. *Subspace Identification for Linear Systems: Theory, Implementation, Applications*. Boston: Kluwer Academic Publishers, 1996.

[WS98] Watts, D. J. & Strogatz, S. "Collective Dynamics of 'Small World' Networks". *Nature* **393** (1998), pp. 409–410.

[Xia05] Xiao, L. et al. "A Scheme for Robust Distributed Sensor Fusion Based on Average Consensus". *Information Processing in Sensor Networks, 2005 Fourth International Symposium on*. 2005, pp. 63–70.

[Xu14] Xu, Y. et al. "Aggregation of Graph Models and Markov Chains by Deterministic Annealing". *Automatic Control, IEEE Transactions on* **59**.10 (2014), pp. 2807–2812.

[Zav11] Zavlanos, M. et al. "Graph-theoretic connectivity control of mobile robot networks". *Proceedings of the IEEE* **99**.9 (2011), pp. 1525–1540.

[Zel09] Zelazo, D. "Graph-theoretic methods for the analysis and synthesis of networked dynamic systems". PhD thesis. University of Washington, 2009.

APPENDICES

Appendix A

Proofs

This appendix contains proofs of theorems, lemmas, and propositions that impeded the natural flow of the thesis. The proofs appear in sequential order according to the corresponding statement.

A.1 Proof of Proposition 2.4.1

Let $\|\cdot\|$ denote $\|\cdot\|_\infty$, the max row norm. For A_{11} we have

$$\|A_{11}\| = \|U^T L^E U\| \leq 2\overline{\gamma}^E. \tag{A.1}$$

The remaining norms can be computed using the subgraphs $\mathcal{G}^k$. We let m_i denote the node weight for $i \in \mathcal{V}^k$, a_{ij} denote the edge weight for $i \sim j,\ i,\ j \in \mathcal{V}^k$, and $d_i = \sum_{j\sim i} a_{ij},\ i,\ j \in \mathcal{V}^k$. We first compute two norms that will be used in the remaining derivations:

$$(M^k)^{-1}(F^k)^T\left(F^k(M^k)^{-1}(F^k)^T\right)^{-1} = \begin{bmatrix} 1-\frac{m_1}{\mathrm{tr}(M^k)} & \frac{-m_2}{\mathrm{tr}(M^k)} & \cdots & \frac{-m_{n_k-1}}{\mathrm{tr}(M^k)} \\ \frac{-m_1}{\mathrm{tr}(M^k)} & 1-\frac{m_2}{\mathrm{tr}(M^k)} & & \vdots \\ \vdots & \frac{-m_2}{\mathrm{tr}(M^k)} & \ddots & \frac{-m_{n_k-1}}{\mathrm{tr}(M^k)} \\ & \vdots & & 1-\frac{m_{n_k-1}}{\mathrm{tr}(M^k)} \\ \frac{-m_1}{\mathrm{tr}(M^k)} & \frac{-m_2}{\mathrm{tr}(M^k)} & \cdots & \frac{-m_{n_k-1}}{\mathrm{tr}(M^k)} \end{bmatrix} \tag{A.2}$$

$$\Rightarrow \|M^{-1}F^T(FM^{-1}F^T)^{-1}\| \leq 2, \tag{A.3}$$

$$F^k(M^k)^{-1}L^k = \begin{bmatrix} \frac{d_1}{m_1}+\frac{a_{1n_k}}{m_{n_k}} & -\frac{a_{12}}{m_1}+\frac{a_{2n_k}}{m_{n_k}} & \cdots -\frac{a_{1(n_k-1)}}{m_1}+\frac{a_{(n_k-1)n_k}}{m_{n_k}} & -\frac{a_{1n_k}}{m_1}-\frac{d_{n_k}}{m_{n_k}} \\ -\frac{a_{12}}{m_2}+\frac{a_{1n_k}}{m_{n_k}} & \frac{d_2}{m_2}+\frac{a_{2n_k}}{m_{n_k}} & \cdots -\frac{a_{2(n_k-1)}}{m_2}+\frac{a_{(n_k-1)n_k}}{m_{n_k}} & -\frac{a_{2n_k}}{m_2}-\frac{d_{n_k}}{m_{n_k}} \\ \vdots & & \ddots & \vdots \\ -\frac{a_{1(n_k-1)}}{m_{n_k-1}}+\frac{a_{1n_k}}{m_{n_k}} & -\frac{a_{2(n_k-1)}}{m_{n_k-1}}+\frac{a_{2n_k}}{m_{n_k}} \cdots & \frac{d_{n_k-1}}{m_{n_k-1}}+\frac{a_{(n_k-1)n_k}}{m_{n_k}} & -\frac{a_{(n_k-1)n_k}}{m_{n_k-1}}-\frac{d_{n_k}}{m_{n_k}} \end{bmatrix} \tag{A.10}$$

$$F^k(M^k)^{-1} = \begin{bmatrix} \frac{1}{m_1} & 0 & \cdots & 0 & -\frac{1}{m_n} \\ 0 & \frac{1}{m_2} & \cdots & 0 & -\frac{1}{m_n} \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & 0 & \frac{1}{m_{n-1}} & -\frac{1}{m_n} \end{bmatrix} \tag{A.4}$$

$$\Rightarrow \|FM^{-1}\| \leq 2/\underline{m}. \tag{A.5}$$

For A_{12}, using (A.3) and $\|U^T L^E\| \leq 2\overline{\gamma}^E$, we have

$$\|A_{12}\| \leq \|U^T L^E\| \|M^{-1}F^T(FM^{-1}F^T)^{-1}\| \leq 4\overline{\gamma}^E. \tag{A.6}$$

For A_{21}, using (A.5) and $\|L^E U\| \leq 2\overline{d}^E$, we have

$$\|A_{21}\| \leq \|FM^{-1}\| \|L^E U\| \leq 4\overline{d}^E/\underline{m}. \tag{A.7}$$

Finally, we split $A_{22} = FM^{-1}(L^I + L^E)M^{-1}F^T(FM^{-1}F^T)^{-1}$ into external and internal components. The upper bound on the external part can be computed using (A.3) and (A.5) as

$$\|FM^{-1}L^E M^{-1}F^T(FM^{-1}F^T)^{-1}\| \leq \|FM^{-1}\| \|L^E\| \|M^{-1}F^T(FM^{-1}F^T)^{-1}\| \leq 8\overline{d}^E/\underline{m}. \tag{A.8}$$

We next lower bound the internal part, which is block diagonal. Each block is of the form

$$L^k = \begin{bmatrix} d_1 & -a_{12} & \cdots & -a_{1(n_k-1)} & -a_{1n_k} \\ -a_{12} & d_2 & \cdots & -a_{2(n_k-1)} & -a_{2n_k} \\ \vdots & & \ddots & & \vdots \\ -a_{1(n_k-1)} & -a_{2(n_k-1)} & \cdots & d_{n_k-1} & -a_{(n_k-1)n_k} \\ -a_{1n_k} & -a_{2n} & \cdots & -a_{(n_k-1)n_k} & d_{n_k} \end{bmatrix}, \tag{A.9}$$

where $a_{ij} = a_{ji}$ and $d_i = \sum_{j=1,\, j\neq i}^{n_k} a_{ij}$, for $i,\ j \in \mathcal{V}^k$. We compute the (i,j)-component of the kth block of the internal part of A_{22}, denoted $[A_{22}^k]_{ij}$, by taking the product of the ith row of (A.10) and j^{th} column of (A.2). It can be shown that $[A_{22}^k]_{ij} = a_{jn_k}/m_{n_k} - a_{ji}/m_i$ when $i \neq j$,

and $[A_{22}^k]_{ii} = a_{in_k}/m_{n_k} + d_i/m_i$. Using these results, we have

$$\|FM^{-1}L^I M^{-1}F^T(FM^{-1}F^T)^{-1}\| = \max_{k,\, i} \left\{ \frac{d_i}{m_i} + \frac{a_{in_k}}{m_{n_k}} + \sum_{j=1,\, j\neq i}^{n_k-1} \left| \frac{a_{in_k}}{m_{n_k}} - \frac{a_{ij}}{m_j} \right| \right\},$$

which can be lower bounded by

$$\|FM^{-1}L^I M^{-1}F^T(FM^{-1}F^T)^{-1}\| \geq \underline{d}^I/\overline{m}. \tag{A.11}$$

Using the bounds in (A.1), (A.6), (A.7), and (A.11), and defining the slow time-scale as $t_s = t(\varepsilon \underline{d}^I)$ we obtain (2.25a)–(2.25b), which is the canonical singular perturbation form for (2.22).

A.2 Proof of Lemma 3.3.2

We solve the GEP of (3.13) to show that the first p eigenvalues of $L_m = M^{-1}L_{\mathcal{G}}$ are simple and that the corresponding eigenvectors follow the same sign pattern as the eigenvectors of L_{m_1}. Let $L_m = M^{-1}(A+E)$, where $A = I_{n_1} \otimes L_2$, $E = L_1 \otimes I_{n_2}$, and $M = M_1 \otimes M_2$. Denote the eigenvalues of A as $0 = \alpha_1 \leq \cdots \leq \alpha_{n_1 \times n_2}$. Denote the set of M-orthogonal eigenvectors corresponding to $\alpha_1 = 0$, with multiplicity n_1, as $Z = [z_1, \ldots, z_{n_1}]$. Notice that $\alpha_1 = 0$ of A has multiplicity n_1. From here, we find a first-order approximation of the n_1 eigenvalues and eigenvectors of L_m that converge to α_1 and $\{z_k\},\ k = 1, \ldots, n_1$, when $E \to 0$. We denote these approximations in matrix form as $\tilde{\Lambda} = \alpha_1 + d\Lambda$, where $d\Lambda = \mathrm{diag}[d\lambda_1, \ldots, d\lambda_{n_1}]$, and $\tilde{V} = Z + dV$, where $dV = [dv_1, \ldots, dv_{n_1}]$. To find Z we construct a set of M-orthonormal vectors $\Xi = [\xi_1, \ldots, \xi_{n1}]$ that span the null space of $(\alpha_1 M - A)$. One choice of vectors is $\Xi = \left(1/\sqrt{\mathrm{tr}(M_2)}\right) M_1^{-1/2} \otimes \mathbf{1}_{n_2}$. Since the columns of Z and the columns of Ξ span the same vector space, every column in Z can be written as a linear combination of the columns in Ξ, i.e. $Z = \Xi B$, where $B \in \mathbb{R}^{n_1 \times n_1}$ is composed of column vectors of real-valued coefficients. Next we solve the GEP of L_m by finding $\tilde{\Lambda}$ and $\tilde{V}$ that satisfy $L\tilde{V} = M\tilde{V}\tilde{\Lambda}$, where $L = A + E$. Expanding L, $\tilde{V}$, and $\tilde{\Lambda}$ yeilds $(A+E)(Z+dV) = M(Z+dV)(\alpha_1 I_n + d\Lambda)$. Distributing terms and equating first-order products results in $EZ + AdV = MZd\Lambda$. Using the substitution $Z = \Xi B$ and pre-multiplying both sides by Ξ^T, this simplifies to $\Xi^T E \Xi B = Bd\Lambda$, since $\Xi^T M \Xi = I_n$ and $\Xi^T A = (A\Xi)^T = (\alpha_1 M \Xi)^T$. Notice that $d\Lambda$ and B are the eigenvalues and eigenvectors, respectively, of $\Xi^T E \Xi$, where

$$\Xi^T E \Xi = \left(\frac{M_1^{-1/2}}{\sqrt{\mathrm{tr}(M_2)}} \otimes \mathbf{1}_{n2} \right)^T (L_1 \otimes I_{n_2}) \left(\frac{M_1^{-1/2}}{\sqrt{\mathrm{tr}(M_2)}} \otimes \mathbf{1}_{n2} \right) = aL_{s1}, \tag{A.12}$$

with $a = n_2/\mathrm{tr}(M_2)$ and L_{s1} is the symmetrized Laplacian of L_1 defined in (3.6). It follows that $d\Lambda = a\,\mathrm{diag}[\bar{\lambda}_1, \ldots, \bar{\lambda}_{n_1}]$, and by Lemma 3.1.2, columns of B have same nodal domains as $\bar{v}_k$. Furthermore, since $\Xi \geq 0$ is block diagonal, the columns of Z also have the same sign pattern as columns $\bar{v}_k$. Now, since $a\bar{\lambda}_p < \mu_2$, we can approximate the first p eigenvalues of $L_{\mathcal{G}}$ as $\tilde{\lambda}_k \approx a\bar{\lambda}_k,\ k = 1, \ldots, p$, noting that these eigenvalues are simple. We complete the proof by find the corresponding eigenvectors $\tilde{v}_2, \ldots, \tilde{v}_p$. We solve for dV by finding vectors that span the nullspace of $(E - Md\Lambda)$. Letting $dv_k = \bar{v}_k \otimes \zeta_k$, where $\zeta_k \in \mathbb{R}^{n_2}$, we have $0 = (E - d\lambda_k M)dv_k$, which can be written as $\left[L_1 \otimes I_{n_2} - a\bar{\lambda}_k(M_1 \otimes M_2)\right](\bar{v}_k \otimes \zeta_k) = L_1\bar{v}_k \otimes (I_{n_2} - aM_2)\zeta_k,\ k = 1, \ldots, n_1$. Since $L_1\bar{v}_k = \bar{\alpha}_k M_1 \neq 0$, $k = 2, \ldots, n_1$, we must find ζ_k such that $(I_{n_2} - aM_2)\zeta_k = 0$, $k = 2, \ldots, n_1$. Since $(I_{n_2} - aM_2)$ is diagonal, we have $[\zeta_k]_j = 1$ if $[M_2]_{jj} = 1/a$, and $[\zeta_k]_j = 0$ otherwise. This implies that dv_k has the same sign pattern (with some extra zeros) as $\bar{v}_k,\ k = 2, \ldots, n_1$. The

form of dv_k and z_k together imply that $\tilde{v}_2, \ldots, \tilde{v}_p$, have the same sign pattern as $\bar{v}_2, \ldots, \bar{v}_p$.

A.3 Proof of Theorem 7.3.1

Without loss of generality we consider a two area network, i.e. $p = 2$. Applying control (7.27) to (7.5) we get

$$\begin{bmatrix} \hat{M}\dot{x}_s(t) \\ \epsilon\dot{x}_f^1(t) \\ \epsilon\dot{x}_f^2(t) \end{bmatrix} = \begin{bmatrix} \Gamma_{11} & \Gamma_{12} & \Gamma_{13} \\ \Gamma_{21} & \Gamma_{22} & \Gamma_{23} \\ \Gamma_{31} & \Gamma_{32} & \Gamma_{33} \end{bmatrix} \begin{bmatrix} x_s(t) \\ \dot{x}_f^1(t) \\ \dot{x}_f^2(t) \end{bmatrix}, \tag{A.13}$$

where

$$\Gamma_{11} = \overline{A}_{11} + \overline{K}_s \qquad \Gamma_{12} = A_{o1} + B_{o1}K_f^1, \qquad \Gamma_{13} = A_{o2} + B_{o2}K_f^2, \tag{A.14}$$

$$\Gamma_{21} = dA_{1o} + B_f^1 K_s^1, \qquad \Gamma_{22} = A_f^1 + B_f^1 K_f^1, \qquad \Gamma_{23} = d\tilde{A}_{12}, \tag{A.15}$$

$$\Gamma_{31} = dA_{2o} + B_f^2 K_s^2, \qquad \Gamma_{32} = d\tilde{A}_{21}, \qquad \Gamma_{33} = A_f^2 + B_f^2 K_f^2. \tag{A.16}$$

We next use a nonsingular transformation [KK78] to decouple the slow and fast components as

$$\begin{bmatrix} \xi_s(t) \\ \xi_f^1(t) \\ \xi_f^2(t) \end{bmatrix} = \begin{bmatrix} I_p - \sum_{i=1}^2 \epsilon E_i N_i & -\epsilon E_1 & -\epsilon E_2 \\ N_1 & I_{(n_1-1)} & 0_{(n_1-1)\times(n_2-1)} \\ N_2 & 0_{(n_2-1)\times(n_1-1)} & I_{(n_2-1)} \end{bmatrix} \begin{bmatrix} x_2(t) \\ \dot{x}_f^1(t) \\ \dot{x}_f^2(t) \end{bmatrix}, \tag{A.17}$$

where $E_i \in \mathbb{R}^{p\times(n_i-1)}$ and $N_i \in \mathbb{R}^{(n_i-1)\times p}$ satisfy the matrix algebraic equations

$$T_1 \triangleq \Gamma_{22}N_1 - \Gamma_{21} + \Gamma_{23}N_2 - \epsilon N_1(\Gamma_{11} - \Gamma_{12}N_1 - \Gamma_{13}N_2) = 0, \tag{A.18a}$$

$$T_2 \triangleq \Gamma_{23}N_2 - \Gamma_{31} + \Gamma_{32}N_1 - \epsilon N_2(\Gamma_{11} - \Gamma_{12}N_1 - \Gamma_{13}N_2) = 0, \tag{A.18b}$$

$$T_3 \triangleq E_1(\Gamma_{22} + \epsilon N_1\Gamma_{12}) + \epsilon E_2 N_2 \Gamma_{12} - \Gamma_{12} + E_2\Gamma_{32} - \epsilon(\Gamma_{11} - \Gamma_{12}N_1 - \Gamma_{13}N_2)E_1 = 0, \tag{A.18c}$$

$$T_1 \triangleq E_2(\Gamma_{33} + \epsilon N_2\Gamma_{13}) + \epsilon E_1 N_1 \Gamma_{13} - \Gamma_{13} + E_1\Gamma_{23} - \epsilon(\Gamma_{11} - \Gamma_{12}N_1 - \Gamma_{13}N_2)E_2 = 0. \tag{A.18d}$$

Substituting (A.17) into (A.13) yields the closed-loop system in the new coordinates, with the form

$$\begin{bmatrix} \hat{M}\dot{\xi}_s(t) \\ \epsilon\dot{\xi}_f^1(t) \\ \epsilon\dot{\xi}_f^2(t) \end{bmatrix} = \begin{bmatrix} \Xi_s & 0_{p\times(n_1-1)} & 0_{p\times(n_2-1)} \\ 0_{(n_1-1)\times p} & \Xi_f^1 & d\tilde{A}_{12} + \epsilon N_1\Gamma_{13} \\ 0_{(n_2-1)\times p} & d\tilde{A}_{21} + \epsilon N_2\Gamma_{12} & \Xi_f^2 \end{bmatrix} \begin{bmatrix} \xi_s(t) \\ \xi_f^1(t) \\ \xi_f^2(t) \end{bmatrix}, \tag{A.19}$$

where $\Xi_s = \Gamma_{11} - \Gamma_{12}N_1 - \Gamma_{12}N_2$, $\Xi_f^1 = \Gamma_{22} + \epsilon N_1\Gamma_{12}$, and $\Xi_f^2 = \Gamma_{33} + \epsilon N_2\Gamma_{13}$.

It can be seen that the operator $T(T_1, T_2, T_3, T_4, \zeta) = [T_1, T_2, T_3, T_4]$ is analytic in its arguments and its partial derivatives with respect to $N_1,\ E_1,\ N_2,\ E_2$ exists and are invertible. Hence, by the implicit function theorem, $N_1,\ E_1,\ N_2,\ E_2$ are analytic in ζ at $\zeta = 0$. Therefore, the matrices of the transformed system (A.13) can be uniformly approximated by the matrices of (7.7). That is,

$$\hat{M}\dot{\xi}_s(t) = \left[\overline{A}_{11} + \overline{K}_s + \mathcal{O}(\|\zeta\|)\right]\xi_s(t), \tag{A.20a}$$

$$\epsilon\dot{\xi}_f^1(t) = \left[(A_f^1 + B_f^1K_f^1) + \mathcal{O}(\|\zeta\|)\right]\xi_f^1(t) + \mathcal{O}(\|\zeta\|)\xi_f^2(t), \tag{A.20b}$$

$$\epsilon\dot{\xi}_f^2(t) = \left[(A_f^2 + B_f^2K_f^2) + \mathcal{O}(\|\zeta\|)\right]\xi_f^2(t) + \mathcal{O}(\|\zeta\|)\xi_f^1(t). \tag{A.20c}$$

We can see from the continuous dependence of (A.20a) on $\|\zeta\|$ that $\xi_s(t) \to x_s(t)$ as $\|\zeta\| \to 0$. Similarly, we can see that in the fast time-scale $t_f = t/\epsilon$ that $\xi_f^i(t_f) \to x_f^i(t),\ i = 1, 2$, uniformly.

Complete controllability of the pairs $(\overline{A}_{11}, I_p)$ and $(A_f^i, B_f^i),\ i = 1, 2$, implies the existence of gain matrices K_f^i and K_s which arbitrarily assign eigenvalues to the matrices $\overline{A}_{11} + \overline{K}_s$ and $A_f^i + B_f^iK_f^i$, respectively [KS72]. Noticing that the closed-loop eigenvalues $\lambda_i^c,\ i = 1, \ldots, n$ are precisely those of system (A.20) and the fact that the eigenvalues of $\overline{A}_{11} + \overline{K}_s$, $A_f^1 + B_f^1K_f^1$, and $A_f^2 + B_f^2K_f^2$ are $\mathcal{O}(\|\zeta\|)$ regular perturbations of those of (A.20). The application of ([Kok99], Ch. 2, Thm. 3.1) completes the proof.

Appendix B

System Identification Algorithms

This appendix briefly reviews two standard system identification algorithms. For a more thorough development of identication algorithms for linear systems, we refer the reader to [VODM96]. For simplicity, we consider a SISO LTI system of the form

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{B.1a}$$

$$y(t) = Cx(t), \tag{B.1b}$$

where $A \in \mathbb{R}^{n\times n},\ B = \mathbf{e}_k \in \mathbb{R}^{n\times 1},\ C = \mathbf{e}_l^T \in \mathbb{R}^{1\times n}$.

B.1 Eigensystem Realization Algorithm

We describe a Hankel-based method here [JP85]. Consider the discrete-time equivalent of (B.1) as

$$x(k+1) = A_d x(k) + B_d u(k) \tag{B.2a}$$

$$y(k) = Cx(k), \tag{B.2b}$$

where $x(k) \in \mathbb{R}^n$, $y(k) \in \mathbb{R}^\ell$ and $u(k) \in \mathbb{R}^{\tilde{m}}$, $A_d = e^{-L_m T},\ B_d = \int_0^T e^{-L_m \tau} B d\tau$, with sampling time T. Our objective is to identify the transfer matrix of (B.2) given $y(k)$. For this, we assume

that each input $u_i(k)$ is a unit impulse. We define the two $(\varsigma \cdot m) \times (\tau \cdot \ell)$ Hankel matrices

$$\mathbf{H}(0) = \begin{bmatrix} y(1) & \cdots & y(\tau) \\ \vdots & \vdots & \vdots \\ y(\varsigma) & \cdots & y(\tau+\varsigma-1) \end{bmatrix} \tag{B.3a}$$

$$\mathbf{H}(1) = \begin{bmatrix} y(2) & \cdots & y(\tau+1) \\ \vdots & \vdots & \vdots \\ y(\varsigma+1) & \cdots & y(\tau+\varsigma) \end{bmatrix}. \tag{B.3b}$$

Since $y(k) = CA_d^{k-1}B_d,\ k = 1,\ \ldots,\ N$, is the kth Markov parameter, (B.3) can be re-written as

$$\mathbf{H}(0) = \begin{bmatrix} CB_d & \cdots & CA_d^{\tau-1}B_d \\ \vdots & \vdots & \vdots \\ CA_d^{\varsigma-1}B_d & \cdots & CA_d^{\varsigma+\tau-2}B_d \end{bmatrix} \tag{B.4a}$$

$$\mathbf{H}(1) = \begin{bmatrix} CA_dB_d & \cdots & CA_d^{\tau}B_d \\ \vdots & \vdots & \vdots \\ CA_d^{\varsigma}B_d & \cdots & CA_d^{\varsigma+\tau-1}B_d \end{bmatrix}. \tag{B.4b}$$

It is clear that $\mathbf{H}(0) = \mathbf{OC}$ and $\mathbf{H}(1) = \mathbf{O}A_d\mathbf{C}$, where $\mathbf{O}$ is the $(\varsigma \cdot \ell) \times n$ observability matrix and $\mathbf{C}$ is the $n \times (\tau \cdot m)$ controllability matrix of (B.2). We compute the $\tilde{n}^{th}$-order realization of (B.2), $\tilde{n} \leq \text{rank}(\mathbf{H}(0))$, by considering the truncated singular value decomposition of $\mathbf{H}(0)$ as $\hat{\mathbf{H}}(0) = \hat{U}\hat{\Sigma}\hat{V}^T$, where $\hat{\Sigma} \in \mathbb{R}^{\tilde{n}\times\tilde{n}}$, $\hat{U} \in \mathbb{R}^{(\varsigma\cdot\ell)\times\tilde{n}}$, and $\hat{V}^T \in \mathbb{R}^{\tilde{n}\times(\tau\cdot m)}$, contain the first $\tilde{n}$ singular values and singular vectors of $\mathbf{H}(0)$. The estimates for $A_d,\ B_d,\ C_d$ can then be written as

$$\hat{A}_d = \hat{\Sigma}^{-1/2}\hat{U}^T\mathbf{H}(1)\hat{V}\hat{\Sigma}^{-1/2}, \tag{B.5a}$$

$$\hat{B}_d = \hat{\Sigma}^{1/2}\hat{V}^T E_m, \tag{B.5b}$$

$$\hat{C} = E_\ell^T\hat{U}\hat{\Sigma}^{1/2}, \tag{B.5c}$$

where $E_m = [I_m\ 0\ \cdots\ 0]^T$ and $E_\ell = [I_\ell\ 0\ \cdots\ 0]^T$ are of conformable size. In a deterministic model $\tilde{n} \leq \text{rank}(\mathbf{H}(0)) = n$. If the measurements are noisy, we must set a cut-off value $\tilde{n} < \text{rank}(\mathbf{H}(0))$ for the system order. Here we require that $\tilde{n} \geq p$.

Next, the three matrices in (B.5) are converted to their respective continuous-time counterparts, $\hat{A},\ \hat{B},\ \hat{C}$, by zero-order hold. We reemphasize that $\hat{A}$, in general, will not be the same as A because of similarity transformation. For the same reason, $\hat{B}$ is not guaranteed to be the same

indicator matrix as B, and therefore cannot be used directly for input localization. Instead, we use $\hat{A}$, $\hat{B}$, $\hat{C}$ to estimate the transfer matrix $\hat{G}(s) = \hat{C}(\lambda I_{\tilde{n}} - \hat{A})\hat{B}$. In particular, for a given input-output pair, we estimate the slow-mode residues from

$$\hat{g}_{ij}^{slow}(s) = \sum_{k=2}^{p} \frac{\hat{R}_{ij}^{k}}{s + \hat{\lambda}_k}, \tag{B.6}$$

where $\hat{R}_{ij}^{k}$ is the kth estimated residue and $\hat{\lambda}_k$ is the kth identified mode.

B.2 Prony's Method

Prony's method assume that timeseries data is available and takes the form

$$y(t) = \sum_{i=1}^{n} R_{kl}^{i} e^{-\lambda_i t}, \tag{B.7}$$

and that discrete samples of the data are available at timesteps of length T. Defining $c_i \triangleq e^{-\lambda_i T}$, the sampled data can be represented as

$$\begin{aligned} y(0) &= R_{kl}^{1} + R_{kl}^{2} + \cdots \\ y(T) &= R_{kl}^{1} c_1 + R_{kl}^{2} c_2 + \cdots \\ y(2T) &= R_{kl}^{1} c_1^2 + R_{kl}^{2} c_2^2 + \cdots \\ &\vdots \end{aligned}$$

It is our goal to estimate the residues R_{kl}^{i} and the eigenvalues $\lambda_i = -\ln c_i$. We solve this problem in two steps: first estimate c_i and λ_i, then use this to determine R_{kl}^{i}. Define a nth-order polynomial with roots c_i as

$$P(z) = \prod_{i=1}^{n} (z - c_i) \tag{B.8}$$

$$= z^n + \beta_{n-1} z^{n-1} + \cdots + \beta_1 z + \beta_0 \tag{B.9}$$

Next notice the recursion in the data (B.7):

$$\begin{bmatrix} y(n) \\ y(n+1) \\ \vdots \\ y(n+\ell) \end{bmatrix} = - \begin{bmatrix} y(n-1) & \cdots & y(0) \\ y(n) & \cdots & y(1) \\ \vdots & & \vdots \\ y(n+\ell-1) & \cdots & y(\ell) \end{bmatrix} \underbrace{\begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{n-1} \end{bmatrix}}_{\beta}. \tag{B.10}$$

Since all of the $y(t)$ values in (B.10) are know, we can solve for β directly (least squares solution is the left psuedo inverse). Once β is known, $P(z)$ is completely defined, and c_i are the roots of this polynomial. Note that one drawback of Prony's method is root finding algorithms are inherently unstable and may be sensitive to noise. Hence, it is important to restrict the rootfinding algorithm to intervals of interest. Finally, we can write the *Vandermonde* equation as

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(K-1) \end{bmatrix} = \begin{bmatrix} 1 & \cdots & 1 \\ c_1 & \cdots & c_n \\ \vdots & & \vdots \\ c_1^{K-1} & \cdots & c_n^{K-1} \end{bmatrix} \begin{bmatrix} R_{kl}^1 \\ \vdots \\ R_{kl}^n \end{bmatrix}. \tag{B.11}$$

The residues $[R_{kl}^1, \ldots, R_{kl}^n]$ can be solved for directly from (B.11).

Appendix C

Graph-theoretic Modeling of Power Systems

C.1 Power System Modeling

Consider a n-bus power system network consisting of n^g synchronous generators and $n - n^g$ load buses. Without loss of generality, let $\{1, \ldots, n^g\}$ denote the generator buses and $\{n^g + 1, \ldots, n\}$ denote the load buses. Let P_i and Q_i denote the total active and reactive power injected into the ith bus, where

$$P_i = \sum_{k=1}^{n} -V_i^2 r_{ik}/z_{ik}^2 - V_i V_k \sin(\theta_{ik} - \alpha_{ik})/z_{ik}, \tag{C.1a}$$

$$Q_i = \sum_{k=1}^{n} -V_i^2 x_{ik}/z_{ik}^2 + V_i V_k \cos(\theta_{ik} - \alpha_{ik})/z_{ik}, \tag{C.1b}$$

where $V_i \angle \theta_i$ is the voltage phasor at the ith bus, $\theta_{ik} = \theta_i - \theta_k$, r_{ik} and x_{ik} are the resistance and reactance of the tie-line joining bus i with bus k, $z_{ik} = \sqrt{r_{ik}^2 + x_{ik}^2}$, and $\alpha_{ik} = \tan^{-1}(r_{ik}/x_{ik})$.

The electromechanical model of the power system can be described as a system of differential-algebraic equations (DAE) [AF03] as

$$\dot{\delta}_i(t) = \omega_s(\omega_i - 1) \tag{C.2a}$$

$$m_i \dot{\omega}_i(t) = P_i^m - P_i^e - d_i(\omega_i - 1), \tag{C.2b}$$

for $i = 1, \ldots, n^g$, and

$$0 = P_i^e + P_i - P_i^l, \qquad 0 = Q_i^e + Q_i - Q_i^l, \tag{C.3a}$$

$$0 = P_k - P_k^l, \qquad 0 = Q_k - Q_k^l, \tag{C.3b}$$

for $i = 1, \ldots, n^g$ and $k = n^g + 1, \ldots, n$. In (C.2), $\delta_i(t)$, $\omega_i(t)$, m_i, and d_i denote the internal rotor angle, speed, inertia, and damping of the ith generator, while P_i^m denotes the mechanical power input, P_i^e and Q_i^e denote the active and reactive electrical power produced by the ith generator. In (C.3), P_k^l and Q_k^l denote the active and reactive power consumed by the load at the kth bus. The DAE (C.2) can be converted into a system of purely differential equations via Kron reduction [DB13], resulting in a fully connected graph $\mathcal{G}^k$ with n^g nodes. Neglecting line losses, its small-signal model linearized around $(\delta_0, \mathbf{1}_{n^g})$, following a disturbance in the network, can be written as

$$\left[\begin{array}{c} \Delta\dot{\delta} \\ \hline \Delta\dot{\omega} \end{array}\right] = \underbrace{\left[\begin{array}{c|c} 0_{n\times n} & \omega_s I_{n\times n} \\ \hline -L_m & -M^{-1}D \end{array}\right]}_{A} \left[\begin{array}{c} \Delta\delta \\ \hline \Delta\omega \end{array}\right] + \underbrace{\left[\begin{array}{c} 0 \\ \hline M^{-1}e_l \end{array}\right]}_{B} u(t), \tag{C.4}$$

where $\Delta\delta = [\Delta\delta_1, \ldots, \Delta\delta_{n^g}]^T$, $\Delta\omega = [\Delta\omega_1, \ldots, \Delta\omega_{n^g}]^T$, $M = \mathrm{diag}[m_1, \ldots, m_{n^g}]$, $D = \mathrm{diag}[d_1, \ldots, d_{n^g}]$, and L_m is the asymmetric weighted Laplacian of the complete graph $\mathcal{G}^K$ with elements

$$[L_m]_{ki} = \begin{cases} -a_{ki}/m_k & \text{if } k \neq i, \\ \sum_{\kappa\neq k} a_{k\kappa}/m_k & \text{if } k = i. \end{cases} \tag{C.5}$$

The edge weights of $\mathcal{G}^K$ are given as

$$a_{ki} = \frac{E_k E_i}{x_{ki}} \cos\left(\delta_{k0} - \delta_{i0}\right) \tag{C.6}$$

for all $k \neq i$, where voltage at the ith and k^{th} generator E_i and E_k, respectively, are taken to be constant at this time scale. The vertex set of $\mathcal{G}^K$ is the set of generator buses $\mathcal{V} = \{1, \ldots, n^g\}$.