University of California
Santa Barbara

Control and Estimation in Network Systems

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Electrical and Computer Engineering

by

Kevin D. Smith

Committee in charge:

Professor Francesco Bullo, Chair
Professor Mahnoosh Alizadeh
Professor João Hespanha
Professor Ambuj Singh

December 2022

The Dissertation of Kevin D. Smith is approved.

Professor Mahnoosh Alizadeh

Professor João Hespanha

Professor Ambuj Singh

Professor Francesco Bullo, Committee Chair

December 2022

Control and Estimation in Network Systems

by

Kevin D. Smith

To my grandfather, Dr. Vince Meyer.

Acknowledgements

I would first like to thank Francesco Bullo, my advisor. Francesco's guidance has been invaluable to my development as a researcher, and I am grateful for his encouragement and advice throughout my PhD program. I hope to carry forward his high standards for quality, passion for research, and kindness toward students in my own career.

In addition to Francesco, I am indebted to my collaborators Giulia De Pasquale, Elizabeth Huang, Saber Jafarpour, Francesco Seccamonte, and Ananthram Swami, whose insights and expertise have provided immeasurable benefit to my research. Working with these talented mathematicians and scientists has been a highlight of my time at UCSB.

I would also like to acknowledge my lab mates for their support and productive conversations: Veronica Centorrino, Pedro Cisneros-Velarde, Sasha Davydov, Robin Delabays, Gilberto Diaz-Garcia, Xiaoming Duan, Anand Gokhale, Sean Jaffe, Yohan John, Shadi Mohagheghi, and Dario Paccagnan. Beyond Francesco's group, I am grateful to have been a part of UCSB's control's community and for the camaraderie of my fellow graduate students in the ECE, ME, and CS departments.

I am thankful to my committee members, Mahnoosh Alizadeh, João Hespanha, and Ambuj Singh, for their constructive feedback on my dissertation research. Special thanks are also due to Val De Veyra and Tim Robinson for their work behind the scenes.

Finally, I would like to thank my family: my parents, for impressing on me the value of education and working tirelessly to support my academic pursuits; and my grandfather, who was the source of my interests in science and engineering. In a way, this thesis is the culmination of their efforts to sustain my curiosity. I am grateful for the support of my family during my graduate studies, especially to my wife, Hannah.

My research was supported by the National Science Foundation award #1258507, by the US Defense Threat Reduction Agency grant HDTRA1-19-1-0017, by the US Army Engineer Research and Development Center grant W912HZ-22-2-0010, and by the Air Force Office of Scientific Research projects A9550-22-1-0059 and FA9550-21-1-0203.

Curriculum Vitæ

Kevin D. Smith

Education

2022	**Ph.D., Electrical and Computer Engineering** University of California, Santa Barbara
2019	**M.S., Electrical and Computer Engineering** University of California, Santa Barbara
2017	**B.S., Physics** Harvey Mudd College

Research Experience

2019–2022	**Graduate Student Researcher** University of California, Santa Barbara
2017–2020	**Ph.D. Intern** Pacific Northwest National Laboratory
2017–2019	**N.S.F. IGERT Trainee** University of California, Santa Barbara
Summer 2016	**Technical Intern** Pacific Northwest National Laboratory
2015–2016	**Undergraduate Researcher** Harvey Mudd College

Awards

Fall 2022	**NeurIPS Scholar Award** 36th Conference on Neural Information Processing Systems
2017–2019	**NSF IGERT Traineeship** University of California, Santa Barbara

Professional Service

Reviewer	*IEEE Control Systems Letters (L-CSS)* *IEEE/ACS Transactions on Networking (TON)* *IEEE Transactions on Circuits and Systems (TCAS-I)* *IEEE Conference on Decision and Control (CDC)* *American Control Conference (ACC)* *Hawaii International Conference on System Sciences (HICSS)*

Publications

1. G. De Pasquale, K. D. Smith, F. Bullo, and M. E. Valcher. Dual seminorms, ergodic coefficients, and semicontraction theory. *Technical Report*, 2022. doi:10.48550/arXiv.2201.03103
2. K. D. Smith and F. Bullo. Contractivity of the method of successive approximations for optimal control. *IEEE Control Systems Letters*, November 2022. To appear. doi:10.48550/arXiv.2209.10046
3. K. D. Smith, F. Seccamonte, A. Swami, and F. Bullo. Physics-informed implicit representations of equilibrium network flows. In *Advances in Neural Information Processing Systems*, 2022. To appear
4. K. D. Smith and F. Bullo. Convex optimization of the basic reproduction number. *IEEE Transactions on Automatic Control*, 2022. To appear. URL: https://arxiv.org/abs/2109.07643
5. K. D. Smith, S. Jafarpour, A. Swami, and F. Bullo. Topology inference with multivariate cumulants: The Möbius inference algorithm. *IEEE/ACM Transactions on Networking*, 30(5):2102–2116, 2022. doi:10.1109/TNET.2022.3164336
6. S. Jafarpour, E. Y. Huang, K. D. Smith, and F. Bullo. Flow and elastic networks on the n-torus: Geometry, analysis and computation. *SIAM Review*, 64(1):59–104, 2022. doi:10.1137/18M1242056
7. K. D. Smith, S. Jafarpour, and F. Bullo. Transient stability of droop-controlled inverter networks with operating constraints. *IEEE Transactions on Automatic Control*, 67(2):633–645, 2022. doi:10.1109/TAC.2021.3053552
8. K. D. Smith. A tutorial on multivariate k-statistics and their computation, 2020. URL: http://arxiv.org/pdf/2005.08373
9. K. D. Smith and K. Studarus. Limited-knowledge economic dispatch prediction using bayesian averaging of single-node models. In *2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*. IEEE, 2018. doi:10.1109/PMAPS.2018.8440564
10. K. D. Smith, S. C. Hsiung, C. White, C. G. Lowe, and C. M. Clark. Stochastic modeling and control for tracking the periodic movement of marine animals via AUVs. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3101–3107. IEEE, 2016. doi:10.1109/IROS.2016.7759480

Abstract

Control and Estimation in Network Systems

by

Kevin D. Smith

Networks are ubiquitous in natural and engineered systems, from critical infrastructures (like power grids and water distribution systems) to contact networks in epidemiological models. Managing these systems requires a broad array of tools to monitor the configuration and state of the network, identify optimal operating points, and design controls. This thesis examines a collection of topics broadly related to this theme.

In Part 1, we consider problems related to control and optimization in network systems. Chapter 1 studies the problem of safety-critical control in networks of grid-forming inverters. Coupling a physically-meaningful Lyapunov-like function with an optimization approach to identifying forward-invariant sets, we propose a method to certify that a post-fault trajectory achieves frequency synchronization while respecting safety constraints. In Chapter 2, we consider the network resource allocation problem of optimally distributing resources to mitigate the spread of an epidemic. We propose a convex optimization framework for minimizing the basic reproduction number for general compartmental epidemiological models. Chapter 3 addresses optimal control in infinitesimally contracting systems. We provide new convergence criteria for a common indirect optimal control algorithm, and we establish the uniqueness of the optimal control in the limits of large contraction rates and short time horizons.

In Part 2, we use tools from statistical inference and machine learning to solve estimation problems in network systems. Chapter 4 examines the problem of inferring routing topologies from endpoint data in communication networks. Extending a technique called network tomography to use higher-order statistics, and using Möbius inversion to disentangle the interactions between different network paths that are reflected in these statistics, we are able to estimate routing matrices without the cooperation of intermediate routers. Finally, in Chapter 5, we use machine learning to predict edge flows. We propose an implicit neural network that incorporates two fundamental physical principles to estimate flows on unlabeled edges, and we provide a contraction mapping to evaluate the model and backpropagate loss gradients.

Contents

Part I

Control

Chapter 1

Safety-Critical Control of Grid-Forming Inverter Networks

This chapter was first published in *IEEE Transactions on Automatic Control* [119].[1]

Due to the rise of distributed energy resources, the control of networks of grid-forming inverters is now a pressing issue for power system operation. Droop control is a popular control strategy in the literature for frequency control of these inverters. In this chapter, we analyze transient stability in droop-controlled inverter networks that are subject to multiple operating constraints. Using a physically-meaningful Lyapunov-like function, we provide two sets of criteria (one mathematical and one computational) to certify that a post-fault trajectory achieves frequency synchronization while respecting safety constraints. We show how to obtain less-conservative transient stability conditions by incorporating information from loop flows, i.e., net flows of active power around cycles in the network. Finally, we use these conditions to quantify the scale of parameter disturbances to which the network is robust. We illustrate our results with numerical

[1] ©2021 IEEE. Reprinted, with permission, from Kevin D. Smith, Saber Jafarpour, and Francesco Bullo, *Transient Stability of Droop-Controlled Inverter Networks With Operating Constraints*, January 2021.

case studies of the IEEE 24-bus system.

1.1 Introduction

Transient stability is a power systems problem of both practical importance and theoretical interest. The goal of transient stability analysis is to determine whether or not the system will return to a stable, frequency-synchronized operating point after a large disturbance. Transients are difficult to analyze: the governing differential equations are nonlinear, and linearization techniques are not useful for large-scale disturbances. Therefore, system operators typically rely on numerical simulation [53, Chapter 9.3] to study system behavior. Simulation is an effective tool for analyzing individual disturbance scenarios, but it has limitations. Simulating a comprehensive set of disturbances is computationally expensive, and it does not establish rigorous guarantees.

Direct methods of transient stability analysis address these limitations by establishing theoretical guarantees on transient behavior. Direct methods are not a substitute for simulation in real-world power system operation, since they rely on low-order, theoretically-tractable models. Instead, they provide significant theoretical insight into these simplified systems. Classical works on using Lyapunov-like methods to study transient stability include [6, 129, 26]. More recently, [132, 133] used set-theoretic control techniques to establish regions of attraction for the coupled swing equations. Direct methods are highly model-specific and provide conservative guarantees, so this topic is still the subject of active research.

Historically, the literature on direct methods has focused on networks of high-inertia synchronous generators. But the rise of distributed energy resources has sparked a growing interest in the stability of low-inertia inverter networks, particularly microgrids. Inertia is both a blessing and a curse from a control perspective—the same inertia that makes the system robust to disturbances also makes the system respond sluggishly to control inputs. A suitable fast-acting controller can make a low-inertia inverter network highly robust. Two broad classes of inverter controllers have emerged to exploit this low inertia: *grid-following* controllers, in which the inverter acts as a current source to track the local voltage signal; and *grid-forming* controllers, in which the inverter acts as a voltage source to stabilize voltage frequencies throughout the network. Both of these frameworks involve new models and require fresh approaches to direct transient stability analysis.

One of the most popular approaches to grid-forming control is *proportional droop control*, in which local voltage frequencies are modulated in proportion to the power drawn from neighboring buses. Recent work [112, 3, 143] has studied the dynamics of droop-controlled microgrids (DCMGs) via the inhomogeneous Kuramoto model. Under certain assumptions, equilibrium points of the Kuramoto model correspond to frequency-synchronized operating points of the DCMG, and regions of attraction around these equilibria provide a rigorous way to assess how robust DCMG operating points are to disturbances. Some progress has been made on estimating these regions of attraction [40], but these closed-form estimates tend to be very conservative and require stringent regularity assumptions on the topology or system parameters.

Another limitation of the literature is that few bounds on the transients are available. To a system operator, a guarantee of frequency synchronization alone is not satisfying, if the resulting transient will violate operating constraints (like constraints on line flows and nodal power injections). Recent work has begun to address transient stability in conjunction with other engineering constraints [82, 81]. If direct methods of transient stability are to provide more insight into the operation of DCMGs, then less-conservative regions of attraction, as well as bounds on quantities of engineering significance, are needed. This chapter addresses these two needs.

Contributions Our first contribution is to extend the transient stability problem. In addition to the classical notion of transient stability (asymptotic frequency synchronization), we impose five "desired properties" of transients, so as to enforce operating constraints on nodal frequencies, power flows across transmission lines, nodal power injections, nodal ramping rates, and reserves of stored energy.

Our second contribution is to provide two sufficient conditions for when a trajectory of a DCMG will exhibit transient stability and the five desired properties. Both certificates require only two pieces of information from from the initial condition (nodal frequencies and line angle differences) instead of the full (and harder to measure) vector of voltage angles. The first certificate can be viewed as a DCMG-specific form of Nagumo's theorem, and it is intended as a theoretical basis for transient-stability-certifying algorithms. The second certificate, which consists of a tractable mixed-integer linear program (MILP), is built on top of the first certificate. These theoretical results use a physically-meaningful Lyapunov function called the "maximum frequency deviation," which (to our knowledge) has not been used before to study power systems.

Our third contribution is to improve these two certificates using the winding partition of the n-torus. We introduced the winding partition in [68] to localize the multiple equilibrium points of network flows on the n-torus. This chapter provides the first application of the winding partition to analyzing system dynamics (in contrast to its previous applications to statics problems). We show how to incorporate the "winding vector" of the initial condition (a quantity closely related to flows of active power around cycles in the network) into the two certificates, resulting in less-conservative conditions for transient stability and the other desired properties.

As a fourth contribution, we use our transient stability conditions to quantify the size of parameter disturbances with respect to which the DCMG is robust. We define a single number that quantifies the "size" of an arbitrary change in model parameters, and we compute a critical threshold such that post-fault transient stability is guaranteed in any disturbance “smaller” than the threshold. We examine particular disturbance modes, including changes in nominal power injections, voltage magnitudes, and branch admittance magnitudes. We illustrate all of these results numerically, using the IEEE 24-bus system as a case study.

Organization The next four sections are organized around our four main contributions. After introducing our notation, model, and problem statement, Section 1.2 states the extended transient stability problem in Definition 1.1, formally introducing transient stability and the five desired properties of the transient. Section 1.3 presents our two certificates in Theorem 1.4 and Theorem 1.6, respectively. We review the winding partition in Section 1.4 and improve both certificates by incorporating the winding vector in Theorem 1.9, and we show that these improved certificates are less conservative (Theorem 1.10). Finally, Section 1.5 uses the stability certificates to study how robust a DCMG is to changes in parameters, and it presents our numerical case studies.

1.2 Preliminaries and Problem Statement

1.2.1 Notation

The Circle and n-Torus Let $\mathbb{S}$ be the circle, i.e., the set of phases or angles. For every pair of angles $\alpha, \beta \in \mathbb{S}$, we use $|\alpha - \beta|$ to denote the geodesic distance between them. The counterclockwise difference between two angles is the map $d_{\mathrm{cc}} : \mathbb{S} \times \mathbb{S} \to [-\pi, \pi)$, where

$$d_{\mathrm{cc}}(\alpha, \beta) = \begin{cases} |\alpha - \beta|, & \text{c.c. arc from } \alpha \text{ to } \beta \text{ shorter than } \pi \\ -|\alpha - \beta|, & \text{otherwise} \end{cases}$$

In other words, we consider the clockwise and counterclockwise arcs from α to β. If the counterclockwise arc is shorter, then $d_{\text{cc}}(\alpha, \beta)$ is the length of that arc. Otherwise, $d_{\text{cc}}(\alpha, \beta)$ is the negated length of the clockwise arc. The n-torus, denoted $\mathbb{T}^n$, is the product of n circles.

Sets Given any set S within $\mathbb{R}^n$ or $\mathbb{T}^n$, we refer to the interior of the set by $\operatorname{int}(S)$, the closure by $\operatorname{cl}(S)$, and the boundary by $\partial S = \operatorname{cl}(S) \setminus \operatorname{int}(S)$, with respect to the standard topologies on $\mathbb{R}^n$ and $\mathbb{T}^n$. Given a function $V : \mathbb{R}^n \to \mathbb{R}$ and a scalar $c \in \mathbb{R}$, we define a sublevel set

$$V_{<}^{-1}(c) = \{x \in \mathbb{R}^n \mid V(x) < c\}.$$

Linear Algebra The vector $\mathbb{1}_n$ (resp. $\mathbb{0}_n$) is a vector in $\mathbb{R}^n$ with all the entries equal to one (resp. zero). For every $v \in \mathbb{R}^n$, $\operatorname{diag}(v) \in \mathbb{R}^{n \times n}$ is a diagonal matrix with entries $\operatorname{diag}(v)_{ii} = v_i$ for every $i \in \{1, \ldots, n\}$. The ∞-norm of v is $\|v\|_\infty = \max_i |v_i|$, and the 1-norm of v is $\|v\|_1 = \sum_{i=1}^n |v_i|$. We define $v_{\text{sum}} = \sum_{i=1}^n v_i$ and $v_{\min} = \min_i\{v_i\}$. For every $v, w \in \mathbb{R}^n$, we write $v \leq w$ (resp. $v < w$) if $v_i \leq w_i$ (resp. $v_i < w_i$), for every $i \in \{1, \ldots, n\}$. For a matrix $X \in \mathbb{R}^{n \times n}$, the Moore–Penrose pseudoinverse is denoted by $X^\dagger$.

Graph Theory An undirected graph is a pair $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of n nodes, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of m edges. The neighborhood of any node $i \in \mathcal{V}$ is denoted by $\mathcal{N}(i)$. While G is undirected, we may enumerate and assign an arbitrary orientation to each edge $e \in \mathcal{E}$ by labeling one incident node as the "source" $s(e)$ and the other as the "sink" $t(e)$. The incidence matrix of the graph [20, §9.1] is the matrix $B \in \{-1, 0, 1\}^{n \times m}$

with entries

$$B_{i,e} = \begin{cases} +1, & s(e) = i \\ -1, & t(e) = i \\ 0, & \text{otherwise} \end{cases}$$

Graphs and the n-Torus We may assign a phase-valued state to every node in G, so that the full state of the graph is in $\mathbb{T}^n$. Given a state $\theta \in \mathbb{T}^n$, we use the abuse of notation $B^\mathsf{T}\theta$ to represent the vector in $\mathbb{R}^m$ of counterclockwise differences across each edge, i.e., $(B^\mathsf{T}\theta)_e = d_{\mathrm{cc}}(\theta_i, \theta_j)$, where i is the source of e and j is the sink. Furthermore, given any $\gamma \in (0, \pi]^m$, we define the phase-cohesive set as the open set

$$\Delta(\gamma) = \left\{\theta \in \mathbb{T}^n : |B^\mathsf{T}\theta| < \gamma\right\}$$

In contrast to the literature, where $\Delta(\gamma)$ takes a scalar-valued γ, the γ we refer to in this chapter is always a vector, allowing for inhomogeneous phase cohesion.

1.2.2 Model

We consider a DCMG on an undirected topology $G = (\mathcal{V}, \mathcal{E})$, with n buses $\mathcal{V} = \{1, \ldots, n\}$ and m branches (or lines) $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. We assume that G is connected, but otherwise we make no assumptions about its structure; both trees and cyclic graphs are acceptable.

Bus Model Each bus has a complex voltage $E_i e^{j\theta_i}$, where $E_i > 0$ is the voltage magnitude and $\theta_i \in \mathbb{S}$ is the phase. We assume that voltage controllers are operating at a much faster time scale than frequency controllers, so that E_i is constant but θ_i is dynamic. We consider two types of buses: droop-controlled inverters and frequency-dependent loads.

Buses in $\mathcal{V}_I \subset \mathcal{V}$ represent droop-controlled inverters, which produce a controllable voltage signal with constant magnitude E_i and time-varying frequency $\dot{\theta}_i$. These inverters operate according to the frequency droop control law [23, 57]:

$$\dot{\theta}_i = \omega^* - \frac{p_{e,i} - p_i^*}{d_i}, \qquad \forall i \in \mathcal{V}_I \tag{1.1}$$

Here $\dot{\theta}_i$ is the instantaneous AC frequency, ω^* is the nominal frequency (for example, 60 Hz), $p_i^* \geq 0$ is the nominal active power injection, and $d_i^{-1} > 0$ is the droop coefficient. Buses in $\mathcal{V}_L = \mathcal{V} \setminus \mathcal{V}_I$ represent frequency-dependent loads [83, §9.1], where the instantaneous active power injection $p_{e,i}$ is

$$p_{e,i} = p_i^* - d_i(\dot{\theta}_i - \omega^*), \qquad \forall i \in \mathcal{V}_L \tag{1.2}$$

Here $p_i^* \leq 0$ is the nominal active power load, and $d_i > 0$. Note that (1.1) and (1.2) are algebraically equivalent.

Branch Model For each branch $\{i, j\} \in \mathcal{E}$, we assume that the real power flow from node i into the $\{i, j\}$ branch is

$$p_{ij}^{\text{line}} = \tilde{a}_{ij} + a_{ij} \sin(\theta_i - \theta_j - \phi_{ij}) \tag{1.3}$$

where $\tilde{a}_{ij} \in \mathbb{R}$, $a_{ij} \geq 0$ and $\phi_{ij} \in (-\frac{\pi}{2}, \frac{\pi}{2})$ are constants. These constants are not necessarily symmetric (i.e., $\phi_{ij} \neq \phi_{ji}$), so in general $p_{ij} \neq -p_{ji}$.

The AC steady-state active power flow across many types of branches can be written in the form (1.3). Transmission lines, for example, are typically represented by the nominal Π model, which consists of a series admittance $Y_{ij}e^{j\varphi_{ij}}$ that is flanked by two shunt admittances $Y_{ii}e^{j\varphi_{ii}}$ and $Y_{jj}e^{j\varphi_{jj}}$ [83, §6.1]. Active power flow in the nominal Π model is given by (1.3) with $\tilde{a}_{ij} = E_i^2(Y_{ii}\cos(\varphi_{ii}) + Y_{ij}\cos(\varphi_{ij}))$, $a_{ij} = E_iE_jY_{ij}$, and $\phi_{ij} = \varphi_{ij} + \frac{\pi}{2}$. In medium-length and short-length lines, the shunt admittance is typically purely capacitive or altogether negligible, either of which leads to the simplification $\tilde{a}_{ij} = E_i^2 Y_{ij}\cos(\varphi_{ij})$. It is also typical that the series admittance is primarily inductive, so $\phi_{ij} \approx 0$. In the extreme case of lossless lines (with no shunt admittance), the active power flow reduces to the antisymmetric form $p_{ij}^{\text{line}} = a_{ij}\sin(\theta_i - \theta_j)$. For transformers, active power flow in the standard equivalent circuit model can also be written in the form (1.3). We omit the specifications of the parameters for brevity and refer the reader to [83, §6.2].

Dynamics Due to conservation of energy, active power injections at each bus must balance against power outflows:

$$p_{e,i} = \sum_{j\in\mathcal{N}(i)} \tilde{a}_{ij} + a_{ij}\sin(\theta_i - \theta_j - \phi_{ij}), \qquad \forall i \in \mathcal{V}$$

Substituting this expression for $p_{e,i}$ into (1.1) and (1.2) leads to a differential equation in θ that captures the angle dynamics of the grid. We can write these dynamics compactly by defining a constant vector $p \in \mathbb{R}^n$ with entries $p_i = p_i^* + \omega^* d_i - \sum_{j\in\mathcal{N}(i)} \tilde{a}_{ij}$ for each $i \in \mathcal{V}$, as well as a matrix $D = \operatorname{diag}\{d_i,\ i \in \mathcal{V}\}$. Then the system can be written as

$$D\dot{\theta} = f(\theta) \tag{1.4}$$

where $f : \mathbb{T}^n \to \mathbb{R}^n$ is a vector with entries

$$f_i(\theta) = p_i - \sum_{j\in\mathcal{N}(i)} a_{ij}\sin(\theta_i - \theta_j - \phi_{ij}), \qquad \forall i \in \mathcal{V}$$

Equation (1.4) is the model that we study in this chapter. If the sine coefficients are homogeneous and the underlying graph is complete, this model is familiar in the physics community as the Kuramoto-Sakaguchi model [110], which has been used to study synchronization phenomena in coupled oscillator networks [139, 35, 18].

Limitations of the Model Our model is based on several commonly-used simplifying assumptions that should be examined explicitly. Perhaps the most important simplification is that we neglect voltage dynamics and reactive power. This is particularly common in the controls community, and it is often justified by assuming fast-acting voltage controllers [133, 135]. If voltage control fails, possibly due to insufficient reactive power, then unmodeled dynamics of the a_{ij} and $\tilde{a}_{ij}$ parameters may destabilize the system.

Another simplification is our use of steady-state AC models for branches in (1.3), which is very common in analysis of conventional power grids. These models assume sinusoidal nodal voltages at a constant frequency, an assumption that is technically contradicted by the dynamic frequencies in (1.4). But the purpose of this chapter is to find sufficient conditions for “safe” transient stability, and a key aspect of safe power grid operation is a tight tolerance around the nominal frequency, typically under 1%. In other words, the trajectories that we are interested in certifying have only a small variance in frequency. Nonetheless, the effects of transmission line dynamics on inverter-based grids is a subject of recent interest, and we refer the reader to [54] for a rigorous study of this topic.

1.2.3 Problem Statement

Under normal operation, nodal frequencies are synchronized at the nominal frequency ω^*, and power injections p_e are equal to the nominal power injections p^*. But contingencies, like failing transmission lines or a sudden change in power supply or demand, disrupt this equilibrium behavior. Droop control will stabilize the post-fault system about a new equilibrium, provided that this new equilibrium is sufficiently close to the pre-fault state.

This local stability property is well-known and easily verified by eigenvalue analysis of (1.4).

Unfortunately, the dynamics of droop control after larger-scale disturbances are not as well understood, and local stability alone does not inspire confidence in a power system controller. Furthermore, the controller should ensure that the system's critical engineering constraints are satisfied during the transient. In this chapter, in addition to non-local transient stability, we consider five engineering constraints that are important in the context of inverter networks:

Definition 1.1 (Desired Properties)**.** *We define the following six properties that are desirable in a trajectory $\theta(t)$ of (1.4):*

(P1) Transient stability. *Nodal frequencies asymptotically synchronize, i.e.,* $\lim_{t\to\infty} \dot{\theta}(t) = \omega_{\text{syn}}\mathbb{1}_n$ *for some synchronous frequency* $\omega_{\text{syn}} \in \mathbb{R}$*.*

(P2) Frequency constraint. *Nodal frequencies are bounded by* $|\dot{\theta}(t) - \omega^*\mathbb{1}_n| \leq \bar{\delta}$ *for all* $t \geq 0$*, where* $\bar{\delta} \geq \mathbb{0}_n$ *is a vector of frequency tolerances.*

(P3) Angle difference constraint. *Voltage angle differences are bounded by* $|B^\mathsf{T}\theta(t)| \leq \bar{\gamma}$ *for all* $t \geq 0$*, where* $\bar{\gamma} \in (0, \frac{\pi}{2}]^m$ *is a vector of angle difference tolerances.*

(P4) Power constraint. *Power injections are sufficiently close to the nominal injection, i.e.,* $|p_e(t) - p^*| \leq \bar{p}_e$ *for all* $t \geq 0$*, where* $\bar{p}_e \in \mathbb{R}^n_{\geq 0}$ *is a vector of power tolerances.*

(P5) Ramping constraint. *The rate of change in power injections is sufficiently small:* $|\dot{p}_e(t)| \leq \bar{r}_e$ *for all* $t \geq 0$*, where* $\bar{r}_e \in \mathbb{R}^n_{\geq 0}$ *is a vector of ramping tolerances.*

(P6) Energy constraint. *The difference from nominal energy injection is bounded by*

$$\left| \int_0^\infty p_e(t) - p^* \; dt \right| \leq \bar{s}$$

where $\bar{s} \in \mathbb{R}^n_{\geq 0}$ is a vector of nodal capacities to store or dump energy.

Each of these desired properties are necessary for safe operation of the power system. (P1) and (P2) are the standard objectives of primary and secondary frequency control, which keep nodal frequencies close to the rated frequency of grid components. (P3) protects transmission lines from overheating, since larger angle differences lead to larger current flow, and thus, more thermal dissipation. (P4) and (P6) ensure that the power and energy drawn from inverters are within a reasonable range. For example, an inverter powered by solar panels on a sunny afternoon is more flexible in its active power injection (via curtailment) than the same inverter on a cloudy morning. Finally, (P5) ensures that the rate at which power injections fluctuate is within the tolerance of the inverter. The objective of this chapter is to find computationally-tractable sufficient conditions on $\theta(0)$ for each of these six properties.

1.3 Main Theoretical Results

We now proceed with our main results: two sets of sufficient conditions to certify that a trajectory satisfies transient stability and the five desired properties in Definition 1.1.

1.3.1 Lyapunov Function

Our analysis is based on the *frequency deviation vector*, which measures the difference between instantaneous and nominal frequencies at each bus:

$$v(\theta) = \dot{\theta} - \omega^* \mathbb{1}_n = D^{-1} f(\theta) - \omega^* \mathbb{1}_n \tag{1.5}$$

From $v(\theta)$, we define our Lyapunov candidate function, the *maximum frequency deviation*

$$V(\theta) = ||v(\theta)||_\infty$$

If a trajectory $\theta(t)$ is clear from context, we will abuse notation and write $V(t)$ instead of $V(\theta(t))$. We will show that the min-max frequency deviation is non-increasing when voltage angle differences are sufficiently small; exactly how small depends on ϕ_{ij}. For each branch, we define a *critical arc length*

$$\gamma_e^* = \frac{\pi}{2} - \max\left\{|\phi_{ij}|, |\phi_{ji}|\right\}, \qquad \forall e = \{i, j\} \in \mathcal{E}$$

Due to the assumption that $\phi_{ij} \in (-\frac{\pi}{2}, \frac{\pi}{2})$, the critical arc lengths satisfy the bound $\gamma_e^* \in (0, \frac{\pi}{2}]$, and the maximum value of $\frac{\pi}{2}$ is achieved by lossless transmission lines (for which $\phi_{ij} = 0$). We collect the critical arc lengths into a vector $\gamma^* \in \mathbb{R}^m$.

As long as the angle difference across each branch is less than the critical arc length, the maximum frequency deviation is non-increasing:

Lemma 1.2 (Max Frequency Deviation is Non-Increasing)**.** *Let $\theta(t)$ be a trajectory of* (1.4) *such that $\theta(t) \in \Delta(\gamma^*)$ on some interval $t \in [t_0, t_1]$. Then $V(t_1) \leq V(t_0)$.*

The proof of this property is based on the following lemma, which is (to our knowledge) novel:

Lemma 1.3 (Sign-Definiteness of Laplacian Matrices)**.** *Let $L \in \mathbb{R}^{n \times n}$ be a Laplacian matrix. For any $x \in \mathbb{R}^n$, let $I_{\max} = \{i : |x_i| = ||x||_\infty\}$ be the set of nodes with maximal absolute value. Then*

$$\max_{i \in I_{\max}} \{-\operatorname{sgn}(x_i)(Lx)_i\} \leq 0$$

Furthermore, if the digraph corresponding to L is strongly connected, then equality holds *if and only if* $x \in \operatorname{span}(\mathbb{1}_n)$.

Proof of Lemma 1.3. For every $i \in I_{\max}$, we compute

$$\begin{aligned} -\operatorname{sgn}(x_i)(Lx)_i &= \sum_{j\neq i} L_{ij}|x_i| - \sum_{j\neq i} L_{ij}\operatorname{sgn}(x_i)x_j \\ &\leq \sum_{j\neq i} L_{ij}(|x_i| - |x_j|) \end{aligned}$$

The first line follows because L has zero row sums and the second line because off-diagonal entries of L are non-positive. But $i \in I_{\max}$ implies that $|x_i| - |x_j| \geq 0$, so we conclude that $-\operatorname{sgn}(x_i)(Lx)_i \leq 0$. Equality clearly holds in the case where $x \in \operatorname{span}(\mathbb{1}_n)$. Now suppose that $\max_{i\in I_{\max}}\{-\operatorname{sgn}(x_i)(Lx)_i\} = 0$, which implies that $\sum_{j\neq i} L_{ij}(|x_i| - \operatorname{sgn}(x_i)x_j) = 0$ for some particular $i \in I_{\max}$. But each summand is non-positive, so $|x_i| = \operatorname{sgn}(x_i)x_j$ for each j for which $L_{ij} \neq 0$; consequently, $x_i = x_j$ for each out-neighbor j of i. It follows that $j \in I_{\max}$. Extending the same argument to j and all of its neighbors, we see that if a directed path exists from i to any node k, then $x_k = x_i$. But the graph is strongly connected, so we conclude that $x \in \operatorname{span}(\mathbb{1}_n)$. □

Proof of Lemma 1.2. We first observe that $\ddot{\theta} = D^{-1}J(\theta)\dot{\theta}$, where $J(\theta)$ is the Jacobian matrix of $f(\theta)$. For $i \neq j$,

$$J_{ij}(\theta) = \frac{\partial f_i(\theta)}{\partial \theta_j} = \begin{cases} -\cos(\theta_i - \theta_j - \phi_{ij}), & \{i,j\} \in \mathcal{E} \\ 0, & \text{else} \end{cases}$$

If $\theta \in \Delta(\gamma^*)$, then $\cos(\theta_i - \theta_j - \phi_{ij}) > 0$. Furthermore, evaluating the diagonal entries of $J(\theta)$ reveals that the matrix has zero row sums, so $-J(\theta)$ is the Laplacian matrix of a weighted, directed graph whose topology is identical to G (treating each undirected edge in G is a pair of directed edges). Note that this graph is strongly connected.

Let $I_{\max} = \{i : |v_i(t)| = ||v(t)||_\infty\}$ be the set of buses with maximal frequency deviation, so we can write

$$V(t) = \max_{i \in I_{\max}} \{\mathrm{sgn}(v_i(t)) v_i(t)\}$$

Using [20, Lemma 15.16(iii)] to compute the upper right Dini derivative of a pointwise-maximum function, we obtain

$$\begin{aligned} \mathrm{D}^+ V(t) &= \max_{i \in I_{\max}} \left\{ \mathrm{sgn}(v_i(t)) \left(D^{-1} J(\theta) \dot{\theta} \right)_i \right\} \\ &= \max_{i \in I_{\max}} \left\{ \mathrm{sgn}(v_i(t)) \left(D^{-1} J(\theta) v(t) \right)_i \right\} \end{aligned}$$

where the last step follows because $\omega^* \mathbb{1}_n \in \ker(J)$. But $D^{-1}J(\theta)$ is a Laplacian matrix corresponding to a strongly connected graph, so by Lemma 1.3, $D^+V(t) \leq 0$. Lemma 1.2 follows from this bound [20, Lemma 15.16(ii)]. □

In summary, the maximum frequency deviation is positive definite about the subspace of frequency-synchronized states, and it is non-increasing inside of $\Delta(\gamma^*)$.

1.3.2 Set-Theoretic Certificate

We now use the maximum frequency deviation to establish a set-theoretic transient stability and operating constraint certification. Our approach is to construct forward-invariant sets using V, based on the following optimization problem:

Problem 1.1 (Min-Max Frequency Deviation)**.** *Let $S \subseteq \Delta(\gamma^*)$. We define $V^*(\partial S)$ to* *be the minimum value of the following:*

$$\begin{aligned} \textit{minimize}: &\quad V(\theta) \\ \textit{variables}: &\quad \theta \in \mathbb{T}^n \\ \textit{subject to}: &\quad \theta \in \partial S \\ &\quad D^{-1} f(\theta) \textit{ is pointed outward from } S \end{aligned}$$

If the problem is infeasible, we define $V^*(\partial S) = +\infty$.

The min-max frequency deviation is the minimum value of $V(\theta)$ along the "outward boundary" of S, i.e., the portion of ∂S where $\dot{\theta}$ is pointed away from the set.

Minimizing a Lyapunov function around a set boundary is a well-established technique for constructing forward-invariant sets—see, for example, Nagumo's 1942 theorem [12, Theorem 4.7]. More recently, [133] applied this technique to a quadratic Lyapunov function for the coupled swing equations. In our case, the min-max frequency deviation is defined so that sets of the form $S \cap V_<^{-1}(V^*(\partial S))$ are forward invariant. This observation, together with the monotonicity of V, leads to the central theorem of the chapter.

Theorem 1.4 (Set-Theoretic Certificate)**.** *Let* $\theta(t)$ *be a trajectory of* (1.4). *Let* $\gamma_0 = |B^\mathsf{T}\theta(0)|$ *and* $\delta_0 = V(\theta(0))$ *denote the initial angle differences and max frequency deviation. If there exist a vector* $\gamma \in [\gamma_0, \gamma^*]$ *and a set* $\Delta(\gamma_0) \subseteq S \subseteq \Delta(\gamma)$ *such that* $\delta_0 < V^*(\partial S)$, *then*

(i) $\theta(t) \in S \cap V_<^{-1}(\delta_0)$ *for all* $t \geq 0$.

(ii) The transient stability property (P1) *is satisfied.*

Further conditions on γ *and* δ_0 *lead to various desirable properties from Definition* 1.1*:*

(iii) The frequency constraint (P2) *is satisfied for each bus* i *if* $\delta_0 \leq \bar{\delta}_i$.

(iv) *The angle difference constraint* (P3) *is satisfied for each line* $\{i, j\}$ *if* $\gamma_{ij} \leq \bar{\gamma}_{ij}$.

(v) *The power constraint* (P4) *is satisfied for each bus* i *if* $\delta_0 \leq \bar{p}_{e,i} d_i^{-1}$.

(vi) *The ramping constraint* (P5) *is satisfied for each bus* i *if*

$$\delta_0 \leq \frac{1}{2}\bar{r}_i \left(\sum_{j \in \mathcal{N}(i)} a_{ij} \right)^{-1}$$

Additionally, in the special case of lossless networks (where $a_{ij} = a_{ji}$ *and* $\phi_{ij} = \phi_{ji} = 0$*), the following is true:*

(vii) *The energy constraint* (P6) *is satisfied for each bus* i *if*

$$\delta_0 \leq \frac{\lambda_2(L)\cos(\gamma_{\max})\bar{s}_i}{d_i/d_{\min}} \left(1 + \frac{1}{2}\log\left(\frac{d_{\text{sum}}}{d_{\min}}\right) \right)^{-1}$$

where $\lambda_2(L)$ *is the smallest non-zero eigenvalue of the Laplacian matrix* $L = B\left(\operatorname{diag}\{a_{ij}\}_{\{i,j\}\in\mathcal{E}}\right)B^\mathsf{T}$.

Proof. To prove statement (i), observe that any trajectory which escapes S must cross through some point on ∂S where $\dot{\theta}$ is pointed outward from S. By definition, $V^*(\partial S) \leq V(\theta)$ at such a point θ. But Lemma 1.2 implies that $V(\theta) \leq V(0)$, which further implies that $\theta(0) \notin V_<^{-1}(\delta_0)$ if the trajectory reaches this point. Forward invariance of $S \cap V_<^{-1}(\delta_0)$ follows by contrapositive. Regarding statement (ii), recall from the proof of Lemma 1.2 that the frequency dynamics can be written $\ddot{\theta} = D^{-1}J(\theta)\dot{\theta}$, where $D^{-1}J(\theta)$ is the negated Laplacian matrix of a strongly connected digraph when $\theta \in \Delta(\gamma^*)$. It follows from [20, Theorem 12.10] that $\dot{\theta}(t)$ converges to a consensus state.

Statements (iii) and (iv) follow trivially from statement (i). Statement (v) follows because droop control relates power injections to frequencies by $p_{e,i} = p_i^* - d_i(\dot{\theta}_i - \omega^*)$ for each $i \in \mathcal{V}$. Therefore $|p_{e,i}(t) - p_i^*| \leq d_i V(t) \leq d_i \delta_0$ for all $t \geq 0$. To prove statement (vi), we observe for each bus i that

$$|\dot{p}_{e,i}| = \left| \sum_{j \in \mathcal{N}(i)} a_{ij} \cos(\theta_i - \theta_j - \phi_{ij})(\dot{\theta}_i - \dot{\theta}_j) \right| \leq 2\delta_0 \sum_{j \in \mathcal{N}(i)} a_{ij}$$

since $\cos(\theta_i - \theta_j - \phi_{ij}) \in (0, 1)$. To prove (vii), observe that

$$\frac{d}{dt} v(\theta)^\mathsf{T} D v(\theta) = 2v(\theta)^\mathsf{T} J(\theta)\dot{\theta} = 2v(\theta)^\mathsf{T} J(\theta) v(\theta)$$

where the last step follows because $\ker(J(\theta)) = \text{span}\{\mathbb{1}_n\}$. Under the lossless assumption, $J(\theta)$ is negated symmetric Laplacian matrix, and the edge weights in the corresponding graph are $a_{ij}\cos(\theta_i - \theta_j)$, which is lower-bounded by $a_{ij}\cos(\gamma_{\max})$. It follows from [20, Lemma 6.9(ii)] that $\lambda(-J(\theta)) \geq \cos(\gamma_{\max})\lambda_2(L)$, so

$$\begin{aligned} \frac{d}{dt} v(\theta)^\mathsf{T} D v(\theta) &\leq -\cos(\gamma_{\max})\lambda_2(L)||v(\theta)||_2^2 \\ &\leq -\cos(\gamma_{\max})\lambda_2(L) d_{\min} \left(v(\theta)^\mathsf{T} D v(\theta) \right) \end{aligned}$$

Therefore $v(\theta)^\mathsf{T} D v(\theta)$ has an exponential upper bound, which decays in time at the rate $\cos(\gamma_{\max})\lambda_2(L) d_{\min}$. The integrand in (P6) can be upper-bounded using both δ_0 and this exponential, yielding the condition in statement (vii). □

Theorem 1.4 simplifies transient analysis in two ways. First, the conditions depend on the quantities γ_0 and δ_0, rather than the full initial state $\theta(0)$. A system operator can measure γ_0 through line flows and δ_0 through nodal frequencies, rather than using state estimation to obtain $\theta(0)$. Second, the theorem recasts transient analysis as the search for a set $S \subseteq \Delta(\gamma^*)$ with a sufficiently large min-max frequency deviation. The remainder of the section examines a computationally-efficient way to search for such a set.

1.3.3 Groundwork for the MILP Certificate

It is impractical to repeatedly evaluate $V^*(\partial S)$ while searching for a set that satisfies Theorem 1.4. Fortunately, we can efficiently compute upper bounds on $V^*(\partial S)$ if we restrict our search to sets of the form $S = \Delta(\gamma)$. We obtain these upper bounds through a series of relaxations to Problem 1.1, and then we use these bounds to establish an easily-computable transient stability certificate. This subsection lays out the first of two relaxations that we make for this certificate.

When $S = \Delta(\gamma)$ for some $\gamma \in (0, \gamma^*]$, Problem 1.1 admits the following relaxation:

Problem 1.2 (Min-Max Frequency Deviation, Lower Bound)**.** *Let* $\gamma \in (0, \gamma^*]$. *We define*

$\widehat{V}(\gamma)$ to be the minimum value of the following:

$$\begin{aligned}
&\min && \|D^{-1}f - \omega^* \mathbb{1}_n\|_\infty && \text{(1.6a)}\\
&w.r.t. && f \in \mathbb{R}^n,\ y \in \mathbb{R}^m,\ \eta^+ \in \mathbb{R}^m,\ \eta^- \in \mathbb{R}^m, &&\\
& && \quad z^+ \in \{0,1\}^m,\ z^- \in \{0,1\}^m &&\\
&s.t. && f = p - B^+A^+\eta^+ - B^-A^-\eta^- && \text{(1.6b)}\\
& && \eta_e^+ = \sin(y_e - \phi_{s(e),t(e)}),\ \forall e \in \mathcal{E} && \text{(1.6c)}\\
& && \eta_e^- = -\sin(y_e + \phi_{t(e),s(e)}),\ \forall e \in \mathcal{E} && \text{(1.6d)}\\
& && |y| \leq \gamma && \text{(1.6e)}\\
& && z_e^+ = 1 \implies y_e = \gamma_e \textit{ and} && \text{(1.6f)}\\
& && \quad d_{s(e)}^{-1} f_{s(e)} - d_{t(e)}^{-1} f_{t(e)} \geq 0,\ \forall e \in \mathcal{E} &&\\
& && z_e^- = 1 \implies y_e = -\gamma_e \textit{ and} && \text{(1.6g)}\\
& && \quad d_{s(e)}^{-1} f_{s(e)} - d_{t(e)}^{-1} f_{t(e)} \leq 0,\ \forall e \in \mathcal{E} &&\\
& && \sum_{e \in \mathcal{E}} z_e^+ + z_e^- = 1 && \text{(1.6h)}
\end{aligned}$$

If the problem is infeasible, we define $\widehat{V}(\gamma) = +\infty$.

Recall that $s(e)$ and $t(e)$ represent the arbitrary "source" and "target" nodes of each $e \in \mathcal{E}$. To express constraint (1.6b) succinctly, we decompose the incidence matrix B into two matrices $B^+, B^- \in \{0,1\}^{n \times m}$, where $(B^+)_{i,e} = 1$ if and only if $s(e) = i$, and $(B^-)_{i,e} = 1$ if and only if $t(e) = i$, so that $B = B^+ - B^-$. We also define two diagonal matrices $A^+ = \operatorname{diag}\{a_{s(e),t(e)},\ \forall e \in \mathcal{E}\}$ and $A^- = \operatorname{diag}\{a_{t(e),s(e)},\ \forall e \in \mathcal{E}\}$. Constraints (1.6f) and (1.6g) are indicator constraints: if $z_e^+ = 1$, then the constraints $y_e = \gamma_e$ and $d_{s(e)}^{-1} f_{s(e)} - d_{t(e)}^{-1} f_{t(e)} \geq 0$ become "active," but these constraints do not apply if $z_e^+ = 0$. Indicator constraints are easily encoded in the MILP framework [13], and many MILP solvers allow indicator constraints to be supplied explicitly.[2]

Problem 1.2 relaxes Problem 1.1 by optimizing the vector of counterclockwise differences $y = B^\mathsf{T}\theta$ directly, instead of optimizing $\theta \in \mathbb{T}^n$. Given this interpretation of y, constraints (1.6b)–(1.6d) ensure that $f = f(\theta)$ and that the cost function (1.6a) is equal to $V(\theta)$. Constraint (1.6e) guarantees that $\theta \in \Delta(\gamma)$, and (1.6f)–(1.6h) ensure that the underlying θ is on the "outward-pointing" boundary of S.

The most important property of Problem 1.2 is that it yields a lower bound to $V^*(\partial\Delta(\gamma))$:

Lemma 1.5 (Problem 1.2 is a Relaxation)**.** *Let $\gamma \in (0, \gamma^*]$, and let $S = \Delta(\gamma)$. The solutions to Problems 1.1 and 1.2 are related by $\widehat{V}(\gamma) \leq V^*(\partial S)$, where equality holds if the underlying graph G is a tree.*

Due to this bound, we can replace the $\delta_0 < V(\Delta(\gamma))$ condition in Theorem 1.4 with the stricter (but computable) condition $\delta_0 < \widehat{V}(\gamma)$.

Theorem 1.6 (Computational Certificate)**.** *Consider a trajectory $\theta(t)$ of (1.4) on any connected graph G. Let $\gamma_0 = |B^\mathsf{T}\theta(0)|$ and $\delta_0 = V(\theta(0))$ denote the initial angle differences and initial max frequency deviation. If there exists a vector $\gamma \in [\gamma_0, \gamma^*]$ such that $\delta_0 < \widehat{V}(\gamma)$, then statements (i)–(vii) from Theorem 1.4 hold, with respect to the set $S = \Delta(\gamma)$.*

Proof. Let $S = \Delta(\gamma)$. By Lemma 1.5, $\delta_0 < \widehat{V}(\gamma) \leq V^*(\partial S)$, so γ and S satisfy the hypothesis of Theorem 1.4. □

Given a particular $\gamma \in [\gamma_0, \gamma^*]$, Theorem 1.6 provides a certificate for transient stability and the other operating constraints in Definition 1.1, using only two properties of the

[2] CPLEX 12.9 supports explicit indicator constraints. Similarly, the Python interface to Gurobi 9 provides the method `Model.addGenConstrIndicator()`. Both links accessed 8/9/2020.

initial condition: the initial angle differences γ_0, and the initial maximal frequency deviation δ_0. Furthermore, Theorem 1.6 replaces Problem 1.1 with Problem 1.2, which is more readily solved by numerical methods.

But two issues still remain. The first problem with Theorem 1.6 is that it is conservative, since Problem 1.2 is a lower bound on Problem 1.1. This bound is only tight in acyclic networks, and the gap between these two problems tends to increase with the number of edges in the graph. In other words, denser graphs lead to more conservative certificates provided by Theorem 1.6 (compared to Theorem 1.4 applied to $S = \Delta(\gamma)$). Closing this gap with additional constraints requires some deeper analysis of the n-torus geometry, which we postpone to Section 1.4.

The second issue is that Problem 1.2 is difficult to solve. It is possible to tackle the problem using nonlinear programming techniques, but it is faster and safer to relax the problem to obtain a lower bound, which we examine next.

1.3.4 MILP Certificate

Problem 1.2 is challenging to solve numerically, since it contains the nonlinear equality constraints (1.6c) and (1.6d). Furthermore, we must be careful about using nonlinear solvers to estimate $\widehat{V}$. If the solver obtains a sub-optimal solution, then this solution may exceed $V^*(\partial\Delta(\gamma))$, thereby invalidating Theorem 1.6. But *any lower bound on $\widehat{V}(\gamma)$ can be used in place of $\widehat{V}(\gamma)$ in Theorem 1.6.* We can get a lower bound—while simultaneously making the problem much easier to solve—by further relaxing Problem 1.2 into a MILP.

The relaxation is conceptually simple; all we need to do is replace (1.6c) and (1.6d) with linear and integer constraints. The general idea is to find a polytope or a union of polytopes that contain the sine curve. Then these bounds can be encoded within the MILP and substituted in for (1.6c) and (1.6d). The process of constructing these

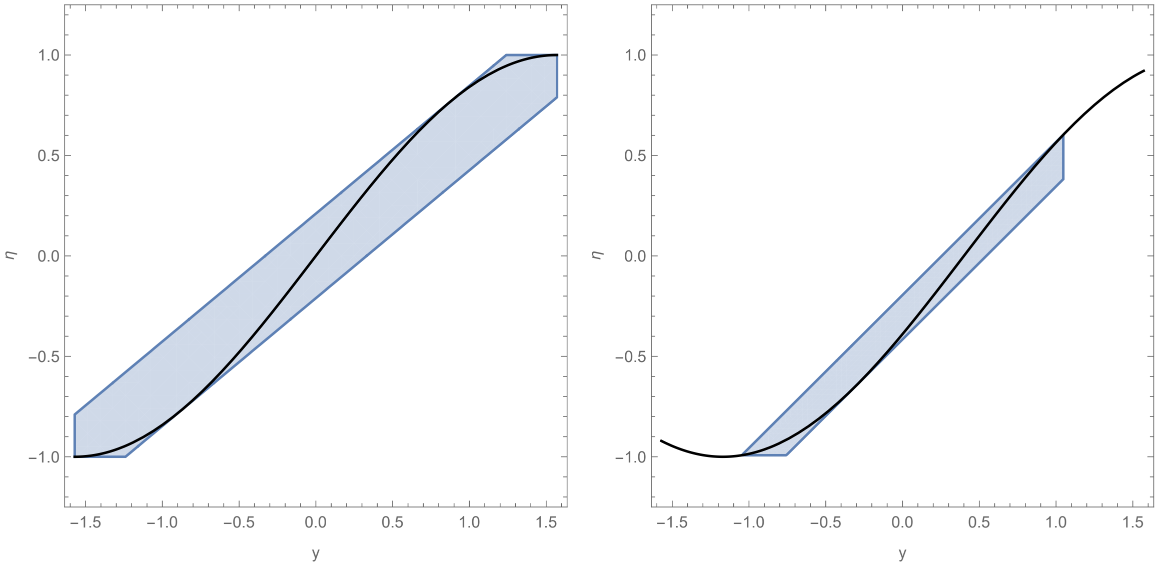

Figure 1.1: Sample relaxations to the constraint $\eta_e^+ = \sin(y_e - \phi_{s(e),t(e)})$ on $y_e \in [-\gamma_e, \gamma_e]$ using bounding polytopes. In the left example, $\phi_{s(e),t(e)} = 0$ and $\gamma_e = \frac{\pi}{2}$. In the right example, $\phi_{s(e),t(e)} = 0.4$ and $\gamma_e = \frac{\pi}{3}$.

polytopes is a messy exercise in elementary geometry, so we do not go into details here. Instead, we present two examples in Figure 1.1, both of which relax the sine constraints with four linear bounds (although tighter and more complicated bounds are clearly possible.) In fact, one can achieve arbitrary precision in the relaxed sine constraints by using piecewise-linear bounds, at the expense of additional binary variables and slower computation.

Replacing (1.6c) and (1.6d) with the polytope relaxation turns Problem 1.2 into a MILP. This MILP can be solved with standard software like Gurobi, CPLEX, or MATLAB. The solution is a lower bound on $\widehat{V}(\gamma)$, which can safely be used in place of $\widehat{V}(\gamma)$ in Theorem 1.6. We also note that this MILP is computationally tractable, since the binary variables essentially split the problem into $2m$ linear programming sub-problems, each corresponding to one of the $2m$ faces of $\Delta(\gamma)$. Informally, Problem 1.2 is no more complex than a collection of $2m$ linear programs.

1.4 Improved Guarantees for Meshed Networks

In the previous section, we found transient guarantees based on two properties of the initial condition, γ_0 and δ_0. But if G is a cyclic topology (i.e., the graph is not a tree), we can make use of an additional property of the initial condition: its *winding vector*, u_0. Winding vectors have recently gained attention in the study of power transmission networks due to their relationship with loop flows [70, 29], i.e., net flows of power around cycles in the network. In [68], we partitioned the n-torus into equivalence classes of winding vectors, and we showed that a wide class of network systems with phase-valued states (including the Kuramoto model) have at most one equilibrium point within each equivalence class. Since winding vectors contain enough information to uniquely characterize equilibrium points of the system, one might also expect them to provide information about the transient behavior.

In this section, we briefly review concepts related to the winding partition. We then apply these concepts to the transient stability problem at hand, using knowledge of the initial winding vector u_0 to obtain less-conservative certificates from Theorem 1.4 and completely close the gap between Problems 1.1 and 1.2. For simplicity, we assume throughout this section that G contains at least one cycle. (Otherwise G is a tree, in which case there is no gap between Problems 1.1 and 1.2 to begin with.)

1.4.1 Winding Partition of the n-Torus

Preliminaries We start with some preliminaries on algebraic graph theory and its application to graph cycles. We refer the reader to [20, §9.3] for a more detailed discussion of these concepts. A *simple cycle* in G is a sequence of consecutive nodes, where the first and last nodes are identical, but all other nodes are distinct. Given a simple cycle $\sigma = (i_1, i_2, \dots, i_{n_\sigma}, i_1)$, the *cycle vector* $v_\sigma \in \mathbb{R}^m$ is defined with respect to the incidence matrix B by

$$(v_\sigma)_e = \begin{cases} +1, & \text{if the edge } e \text{ is traversed positively by } \sigma, \\ -1, & \text{if the edge } e \text{ is traversed negatively by } \sigma, \\ 0, & \text{otherwise.} \end{cases}$$

for each $e \in \mathcal{E}$. More formally, given an adjacent pair of nodes i_j, i_{j+1} in σ, we say that σ traverses the edge $\{i_j, i_{j+1}\}$ *positively* if $B_{i_j,e} = +1$ and $B_{i_{j+1},e} = -1$; otherwise, it traverses the edge *negatively.* The set of cycle vectors for all simple cycles in G span a vector space, called the *cycle space* of G. A set of simple cycles Σ is called a *cycle basis* if the cycle vectors corresponding to elements of Σ are a basis for the cycle space. A cycle basis $\Sigma = \{\sigma_1, \sigma_2, \dots, \sigma_{|\Sigma|}\}$ can be encoded in a cycle-edge incidence matrix $C_\Sigma \in \{-1, 0, 1\}^{|\Sigma| \times m}$, where

$$C_\Sigma = \begin{pmatrix} v_{\sigma_1} & v_{\sigma_2} & \cdots & v_{\sigma_{|\Sigma|}} \end{pmatrix}^\mathsf{T}$$

Clearly $\mathrm{Img}(C_\Sigma^\mathsf{T})$ is the cycle space. Furthermore, because G is a connected graph, the dimension of the cycle space is $|\Sigma| = m - n + 1$.

Winding Vectors and Winding Partition We now review basic definitions regarding winding vectors and the winding partition. The partition divides the n-torus into equivalence classes induced by an underlying graph G. These equivalence classes are defined by how many times the phase differences across basis cycles of G "wind" around the unit circle:

Definition 1.7 (Winding Numbers, Vectors, and Cells)**.** *Let $\theta \in \mathbb{T}^n$. Given any simple* *cycle σ in G with n_σ nodes, the winding number of θ along σ is*

$$w_\sigma(\theta) = \frac{1}{2\pi} \sum_{i=1}^{n_\sigma} d_{\text{cc}}(\theta_i, \theta_{i+1}) \tag{1.7}$$

where the nodes in σ are indexed $\sigma = (1, \dots, n_\sigma, 1)$ and $\theta_{n_\sigma+1} = \theta_1$. Given a cycle basis Σ of G, the winding vector of θ along Σ is the vector

$$w_\Sigma(\theta) = \begin{pmatrix} w_{\sigma_1}(\theta) & w_{\sigma_2}(\theta) & \cdots & w_{\sigma_{|\Sigma|}}(\theta) \end{pmatrix}^\mathsf{T} \tag{1.8}$$

For every winding vector $u \in w_\Sigma(\mathbb{T}^n)$, the u-winding cell is the equivalence class

$$\Omega_u = \{\theta \in \mathbb{T}^n : w_\Sigma(\theta) = u\} \tag{1.9}$$

We note that winding cells were introduced in [68], but winding vectors have been used in the study of power flows since [70]. The reader has likely encountered a similar concept of “winding number” from interpreting Nyquist plots.

Winding vectors are always integer-valued, a property which is analogous to Kirchoff’s voltage law (KVL). For real-valued nodal potentials, KVL guarantees that potential differences sum to zero around any cycle. Similarly, phase differences (in the sense of counter-clockwise arc length) sum to an integer multiple of 2π around any cycle. For example, suppose that G is the triangle graph, consisting of a single cycle $\sigma = (1, 2, 3, 1)$. Let $\theta \in \mathbb{T}^3$. If there is an arc of length π that contains $\theta_1, \theta_2, \theta_3$, then $w_\sigma(\theta) = 0$. Otherwise, $w_\sigma(\theta) = \pm 1$. Figure 1.2 (top) illustrates these three possible winding numbers, based on the configuration of phases around the cycle. Meanwhile, Figure 1.2 (bottom) illustrates Ω_u for each $u = -1, 0, 1$.

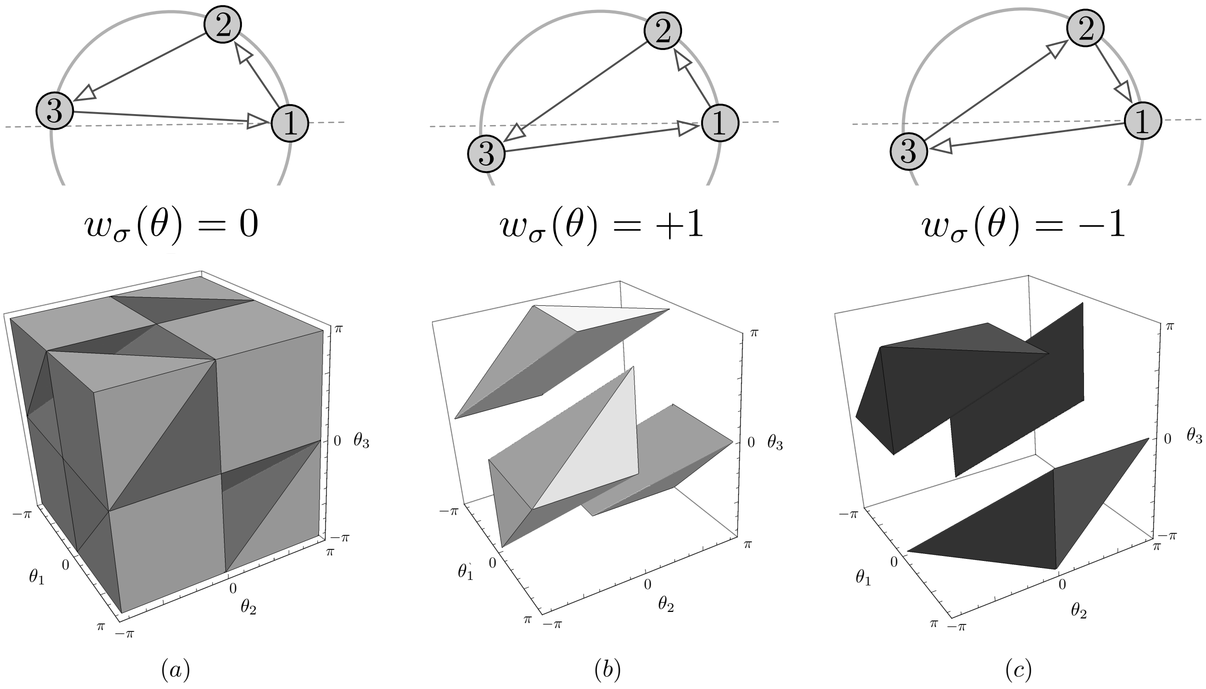

Figure 1.2: Possible winding numbers for any 3-torus state on the 3-cycle (top), and the winding cells in $\mathbb{T}^3$ corresponding to each of these possible winding numbers (bottom).

The cycle basis of a graph is often non-unique, and each valid cycle basis Σ leads to a different definition of the winding vector $w_\Sigma(\theta)$. However, it turns out that the equivalence classes of w_Σ, namely the winding cells, do *not* depend on the particular choice of cycle basis. Given two cycle bases Σ and Σ', every winding cell based on w_Σ is identical to a winding cell based on $w_{\Sigma'}$.

Physical Interpretation of Winding Vectors Winding vectors are closely connected to net flows of active power around cycles in the network. These flows, called loop flows, are of considerable interest to the power systems community because they do not deliver useful power and can jeopardize system stability. For example, flows around the Lake Erie Loop were a major factor in the 2003 Northeast Blackout [55]. Rigorous connections between loop flows and winding vectors are discussed in detail in [70, 29, 68]. We here provide some basic intuition and show how to measure the initial winding vector using line flows, instead of the full state $\theta(0)$.

Consider the active power flows across each line given by (1.3). We assume that these flows are measurable, so that p^{line} is known, even if the state θ is not. If $\theta \in \Delta(\gamma^*)$, then (1.7) can be written

$$w_\sigma(\theta) = \frac{1}{2\pi} \sum_{i=1}^{n_\sigma} \arcsin\left(\frac{p^{\text{line}}_{i,i+1} - \tilde{a}_{i,i+1}}{a_{i,i+1}} \right) + \phi_{i,i+1} \tag{1.10}$$

If we ignore shunt and series losses by setting $\tilde{a}_{i,i+1} = 0$ and $\phi_{i,i+1} = 0$, and we expand the arcsine function about the origin, we obtain

$$w_\sigma(\theta) = \frac{1}{2\pi} \sum_{i=1}^{n_\sigma} \frac{p^{\text{line}}_{i,i+1}}{a_{i,i+1}} + O((p^{\text{line}}_{i,i+1})^3)$$

The quantity $a^{-1}_{i,i+1} p^{\text{line}}_{i,i+1}$ is a normalized line flow, scaled by the capacity of the line. Thus, up to second order, the winding number is a normalized loop flow (at least in the case of short, lossless transmission lines). While somewhat informal, this analysis suggests that winding vectors are a quantized measure of these loop flows. We can also use (1.10) to infer the winding vector from line flow data. Therefore, like the vector of angle differences γ_0, we can identify the initial winding vector u_0 using measurements of line flows instead of the full state $\theta(0)$.

1.4.2 Improved Certificates

We have previously seen (in Theorems 1.4 and 1.6) how to certify transient stability and other desirable properties from the initial condition, using the measurable quantities γ_0 and δ_0 instead of the full state $\theta(0)$. As with γ_0 and δ_0, the initial winding vector $u_0 = w_\Sigma(\theta(0))$ provides additional information that we can exploit to make guarantees about the transient. In the remainder of this section, we will show how to use the winding vector to obtain better certificates out of Theorems 1.4 and 1.6, replacing the antecedents of these theorems with less-conservative conditions.

The new conditions are straightforward to state and prove. We modify Theorem 1.4 to search over sets of the form $\Delta(\gamma_0) \cap \Omega_{u_0} \subseteq S \subseteq \Delta(\gamma) \cap \Omega_{u_0}$ instead of $\Delta(\gamma_0) \subseteq S \subseteq \Delta(\gamma)$. Reducing the lower bound from $\Delta(\gamma_0)$ to $\Delta(\gamma_0) \cap \Omega_{u_0}$ directly incorporates u_0 and results in a larger search space for S, thereby expanding the set of cases that satisfy the antecedent of Theorem 1.4. Shrinking the upper bound from $\Delta(\gamma)$ to $\Delta(\gamma) \cap \Omega_{u_0}$ is not strictly necessary, but as we will see later on, we can always find an optimal S within this smaller upper bound. Similarly, we will modify Theorem 1.6 by adding a constraint to Problem 1.2 that forces the optimum to reside within Ω_{u_0}:

Problem 1.3 (Min-Max Frequency Deviation, Exact)**.** *Let $\gamma \in (0, \gamma^*]$ and $u \in \mathrm{Img}(w_\Sigma)$. We define $\widehat{V}(\gamma, u)$ as the minimum value of Problem 1.2, under the additional constraint $C_\Sigma y = 2\pi u$. If the problem is infeasible, we define $\widehat{V}(\gamma, u) = +\infty$.*

The additional $C_\Sigma y = 2\pi u$ constraint confines the solution to Ω_u, completely closing the gap between Problems 1.1 and 1.2:

Lemma 1.8 (Relations of Minima). *Let $\gamma \in (0, \gamma^*]$, let $u \in \operatorname{Img}(w_\Sigma)$, and let $S = \Delta(\gamma) \cap \Omega_u$. The solutions to Problems 1.1, 1.2, and 1.3 are related by*

$$\widehat{V}(\gamma) \leq \widehat{V}(\gamma, u) = V^*(\partial S).$$

We can now state the new transient stability certificates that account for the initial winding vector.

Theorem 1.9 (Certificates with Winding Vectors). *Consider a trajectory $\theta(t)$ of (1.4), and let Σ be a cycle basis of the underlying graph. Let $\gamma_0 = |B^\mathsf{T}\theta(0)|$, $\delta_0 = V(\theta(0))$, and $u_0 = w_\Sigma(\theta(0))$ denote the initial angle differences, max frequency deviation, and winding vector. Consider the following two conditions:*

(a) There exist a vector $\gamma \in [\gamma_0, \gamma^*]$ and a set $\Delta(\gamma_0) \cap \Omega_{u_0} \subseteq S \subseteq \Delta(\gamma) \cap \Omega_{u_0}$ such that $\delta_0 < V^*(\partial S)$.

(b) There exists a vector $\gamma \in [\gamma_0, \gamma^*]$ such that $\delta_0 < \widehat{V}(\gamma, u_0)$.

If either (a) or (b) are true, then statements (i)–(vii) from Theorem 1.4 hold, with respect to S from condition (a) or $S = \Delta(\gamma) \cap \Omega_{u_0}$ from condition (b).

Proof. The proof that statements (i)–(vii) follow from (a) is identical to the proof of Theorem 1.4, since the new upper and lower bounds on S do not impact the argument for statement (i), and (because S is still contained within $\Delta(\gamma)$) they have no bearing on statements (ii)–(vii). We can use this result from condition (a) to prove that (i)–(vii) follow from condition (b). Let $S = \Delta(\gamma) \cap \Omega_{u_0}$, and observe that $\delta_0 < \widehat{V}(\gamma, u_0) \leq V^*(\partial S)$ due to Lemma 1.8. Then S satisfies (a), so all of the statements hold. □

It is straightforward to show that these new conditions which incorporate u_0 are valid certificates for transient stability, but this is not enough—if we are to go to the trouble of measuring u_0, we would like the assurance that this additional information actually leads to better transient stability certificates. With some simple but careful reasoning about the winding partition, we can see that (a) and (b) are less-conservative versions of the antecedents to Theorems 1.4 and 1.6, respectively:

Theorem 1.10 (Theorem 1.9 is less conservative than Theorems 1.4 and 1.6)**.** *Consider a trajectory $\theta(t)$ of (1.4), let Σ be a cycle basis of the underlying graph, and let $\gamma_0 = |B^\mathsf{T}\theta(0)|$, $\delta_0 = V(\theta(0))$, and $u_0 = w_\Sigma(\theta(0))$. The following are true:*

(i) If the hypothesis of Theorem 1.4 is satisfied, i.e., if there exist a vector $\gamma \in [\gamma_0, \gamma^*]$ and a set $\Delta(\gamma_0) \subseteq S \subseteq \Delta(\gamma)$ such that $\delta_0 < V^*(\partial S)$, then the set $S' = S \cap \Omega_{u_0}$ satisfies condition (a) of Theorem 1.9.

(ii) If the hypothesis of Theorem 1.6 is satisfied, i.e., if there exists a vector $\gamma \in [\gamma_0, \gamma^*]$ such that $\delta_0 < \widehat{V}(\gamma)$, then γ satisfies condition (b) of Theorem 1.9.

Proof. To prove (i), it is sufficient to show that $V^*(\partial S') \geq V^*(\partial S)$, for which it is sufficient to show that $\partial(S \cap \Omega_{u_0}) \subseteq \partial S$. Because the winding cells partition $\mathbb{T}^n$, each of the sets $\Delta(\gamma) \cap \Omega_u$ are disjoint. In fact, because $\gamma < \pi\mathbb{1}_m$, the boundaries of these sets are non-overlapping. Since $S \subseteq \Delta(\gamma)$, we may conclude that ∂S itself is partitioned into non-overlapping pieces $\partial(S \cap \Omega_{u_0})$; hence $\partial(S \cap \Omega_{u_0}) \subseteq \partial S$. Similarly, for (ii) it is sufficient to show that $\widehat{V}(\gamma, u_0) \geq \widehat{V}(\gamma)$, which we have from Lemma 1.8. □

The initial winding vector provides an additional bit of information about the initial state $\theta(0)$, and like the vector of initial angle differences γ_0, the initial winding vector u_0 can be inferred from measurements of active power flows. With knowledge of u_0, we can replace the set-theoretic certificate in Theorem 1.4 and the MILP certificate in Theorem 1.6 with the less-conservative conditions (a) and (b) of Theorem 1.9.

1.5 Quantifying Robustness

An important task in power systems control is understanding how robust an operating point is to disturbances. It is straightforward to study the effects of a *particular* disturbance using simulation, but simulating a comprehensive set of contingencies (or combinations thereof) is time consuming. Our transient stability certificates can aid with this analysis by quantifying the scale of disturbances to which an operating point is robust.

In this section, we consider a DCMG that is operating at a synchronous state θ_0. At time $t = 0$, certain model parameters undergo an instantaneous perturbation—nominal injections change, for example, or a drop in nodal voltages or branch admittances occurs. The initial condition θ_0 is no longer a synchronous state in the "post-fault" model. If the system is sufficiently resilient, then the post-fault transient will settle back down to a synchronous state, and none of the engineering constraints will be violated in the process—but this is not always the case. We will construct a sufficient condition for post-fault transient stability, based on the scale of the perturbations to model parameters.

Numerical Case Study Throughout the section, we will illustrate our results using numerical examples from the IEEE-RTS 24-bus test case [52]. We parameterized (1.4) using branch and bus values from this test case, and we selected the initial voltage angles $\theta_0 \in \mathbb{T}^n$, voltage magnitudes, and nominal power injections by solving for the optimal power flow in MATPOWER. For simplicity, we chose uniform droop coefficients of 10 pu·s and a uniform nominal frequency of 60 s^{-1}. We will refer to this model as the "pre-fault" model. Note that θ_0 is a synchronous equilibrium of the pre-fault model (since it solves the active power flow equations with nominal injections), and the initial winding vector is $u_0 = \mathbb{0}_{11}$ (corresponding to the winding cell with minimal loop flows).

Code The code that we used to generate numerical results in this section is publicly available at https://github.com/KevinDalySmith/DCMG-transient-stability. The optimization problems are implemented using the Python interface to Gurobi 9.0, so a local installation of Gurobi and an active license are needed to run it.

1.5.1 Evaluating Post-Fault Transient Stability

We begin by examining how Theorems 1.6 and 1.9 apply to the problem of quantifying system robustness. Both of these theorems certify transient stability if the post-fault initial condition (i.e., the pre-fault synchronous state) is sufficiently close to a post-fault synchronous state, as measured by the initial max frequency deviation, $\delta_0 = V(\theta_0)$. Then transient stability is certified if there exists any $\gamma \in [\gamma_0, \gamma^*]$ such that $\delta_0 < \widehat{V}(\gamma)$ or $\delta_0 < \widehat{V}(\gamma, u_0)$, as in the following example.

Example 1.11 (Certificates in the 24-Bus System)**.** *To illustrate Theorems 1.6 and 1.9 in the 24-bus system, we randomly select a set of 100 test points $\Gamma \subset [\mathbb{0}_m,\ \gamma^*]$ and evaluate $\widehat{V}(\gamma)$ and $\widehat{V}(\gamma, u_0)$ at each $\gamma \in \Gamma$, with $u_0 = \mathbb{0}_{11}$. Consider an arbitrary initial condition with a maximum angle difference $\bar{\gamma} = ||B^\mathsf{T}\theta(0)||_\infty$ and initial frequency deviation $\delta_0 = V(\theta(0))$. Theorem 1.6 certifies transient stability of the resulting trajectory if*

$$\delta_0 < \max_{\gamma\in\Gamma} \left\{ \widehat{V}(\gamma) : \gamma \geq \bar{\gamma}\mathbb{1}_m \right\}.$$

Under the additional assumption that $w_\Sigma(\theta(0)) = \mathbb{0}_{11}$, Theorem 1.9 certifies transient

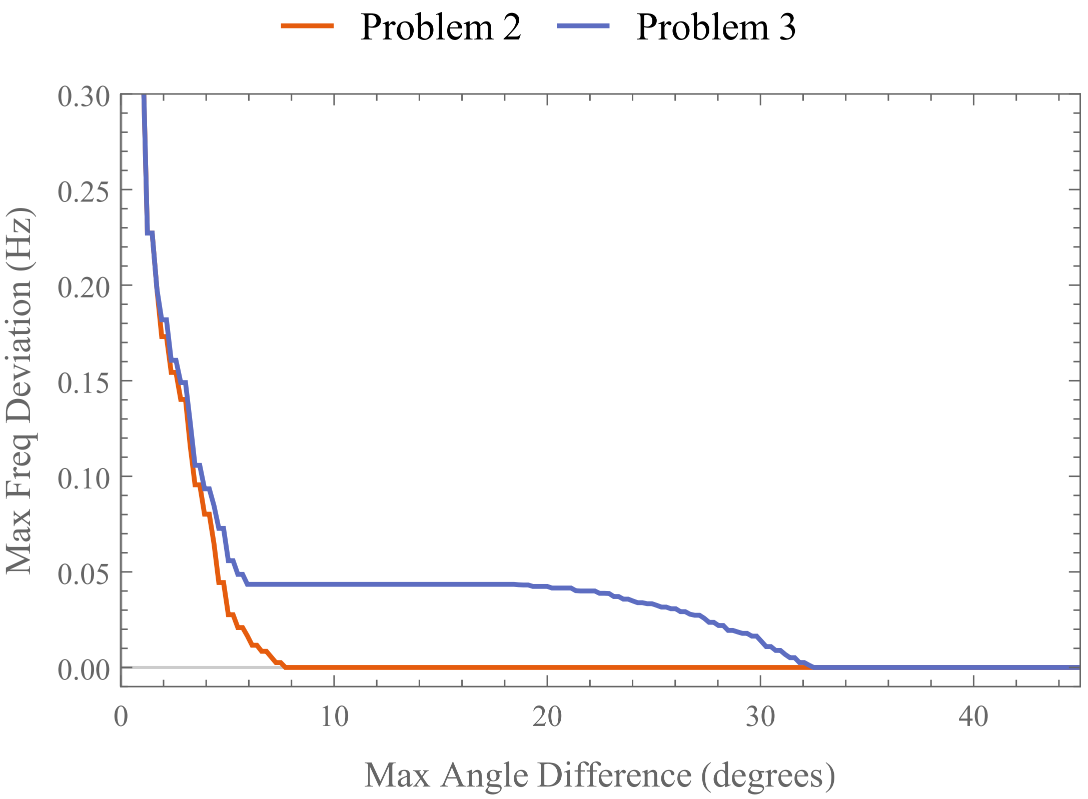

Figure 1.3: Transient stability certification in the IEEE 24-bus test case. The horizontal axis is the max angle difference $||B^{\mathsf{T}}\theta_0||_\infty$ of the initial condition, and the vertical axis is the largest max frequency deviation $V(\theta_0)$ for which transient stability is certified. The lower curve is computed using $\widehat{V}(\gamma)$, and the upper curve is computed using $\widehat{V}(\gamma, \mathbb{0}_{11})$.

stability if

$$\delta_0 < \max_{\gamma \in \Gamma} \left\{ \widehat{V}(\gamma, \mathbb{0}_{11}) : \gamma \geq \bar{\gamma} \mathbb{1}_m \right\}.$$

Figure 1.3 plots both of these conditions. Each curve plots the right-hand side of the preceding inequalities as a function of $\bar{\gamma}$ *(using polytopic relaxations to the sine constraints). For any initial condition corresponding to a point* $(\bar{\gamma}, \delta_0)$ *below the curve, transient stability is certified.*

The curve maximizing $\widehat{V}(\gamma)$ *is significantly lower than the curve maximizing* $\widehat{V}(\gamma, \mathbb{0}_{11})$*, i.e., the transient stability condition from Theorem 1.6 is more conservative than that from Theorem 1.9 (as guaranteed by Theorem 1.10). This plot makes a strong case for incorporating information from the initial winding vector—without it, only very small angle disturbances are certified in the 24-bus system.*

In order to certify transient stability after a fault, we must ensure that the initial post-fault frequency deviation is below the critical threshold. One approach is to follow the procedure of Example 1.11: generate a plot similar to Figure 1.3 using the post-fault parameters, and check whether or not θ_0 corresponds to a point below the curve. But this approach is cumbersome when considering a large number of contingencies, and it offers little advantage over simulation.

A much more efficient approach, similar to that in [133], is to define the "size" of a general disturbance and establish a threshold below which transient stability is certified in all disturbances that are "smaller" than the threshold. A natural way to define the size of a disturbance is to quantify its effect on the frequency deviation vector from (1.5). Suppose that $v : \mathbb{T}^n \to \mathbb{R}^n$ is the frequency deviation vector field defined with the pre-fault model parameters, and similarly, let $\bar{v}$ be the vector field defined with the post-fault parameters. If we can bound the difference $\xi(\theta) = \bar{v}(\theta) - v(\theta)$, then we can bound the solutions to Problems 1.2 and 1.3 after the disturbance based on their solutions before the disturbance:

Theorem 1.12 (Robustness to Parameter Changes)**.** *Consider the model* (1.4)*, and let* $v : \mathbb{T}^n \to \mathbb{R}^n$ *be the associated frequency deviation vector. Let* $\theta_0 \in \mathbb{T}^n$ *be a state for which* $v(\theta_0) = 0$*, i.e., for which all nodal frequencies are identical to* ω^**. After some perturbation in model parameters, suppose that the new frequency deviation vector is given by* $\bar{v}(\theta) = v(\theta) + \xi(\theta)$*, and let* $\gamma_0 = |B^\mathsf{T}\theta_0|$ *and* $u_0 = w_\Sigma(\theta_0)$ *in the post-fault model. If there exists* $\gamma \in [\gamma_0, \gamma^*]$ *such that*

$$||\xi(\theta_0)||_\infty + \max_{\theta\in\partial S} ||\xi(\theta)||_\infty < \min_{\theta\in\partial S} ||v(\theta)||_\infty \tag{1.11}$$

where either $S = \Delta(\gamma)$ *or* $S = \Delta(\gamma) \cap \Omega_{u_0}$*, then the trajectory of the perturbed model starting from* θ_0 *satisfies statements (i)–(vii) from Theorem 1.4 with respect to* S*.*

Proof. Let $\Theta \subseteq \partial S$ be the feasible set of Problem 1.1 evaluated on the perturbed model, i.e., the set of points $\theta \in \partial S$ such that $D^{-1}f(\theta)$ is pointed outward from S. Then the solution to Problem 1.1 (evaluated on the perturbed model) is

$$\begin{aligned} V^*(\partial S) &= \min_{\theta\in\Theta} \{||v(\theta) + \xi(\theta)||_\infty\} \\ &\geq \min_{\theta\in\Theta}\{||v(\theta)||_\infty\} - \max_{\theta\in\Theta}\{||\xi(\theta)||_\infty\} \\ &\geq \min_{\theta\in\partial S}\{||v(\theta)||_\infty\} - \max_{\theta\in\partial S}\{||\xi(\theta)||_\infty\} \end{aligned}$$

Given the initial condition θ_0 to the perturbed model, the initial frequency deviation is $\delta_0 = ||v(\theta_0) + \xi(\theta_0)||_\infty = ||\xi(\theta_0)||_\infty$, so applying (1.11) and the lower bound on $V^*(\partial S)$, we obtain

$$\delta_0 = ||\xi(\theta)||_\infty < \min_{\theta\in\partial S} ||v(\theta)||_\infty - \max_{\theta\in\partial S} ||\xi(\theta)||_\infty \leq V^*(\partial S)$$

Therefore γ and S satisfy the hypothesis of Theorem 1.4 in the perturbed model, and the theorem statements follow. □

Condition (1.11) bounds the scale of the perturbation $\xi(\theta)$. As we will see, in many cases, the left-hand size of the equation is straightforward to compute (or at least upper bound). The right-hand side of (1.11) is *almost* identical to either $\widehat{V}(\gamma)$ or $\widehat{V}(\gamma, u_0)$ (depending on whether S is intersected with the winding cell); the only difference is that the "$D^{-1}f(\theta)$ is pointed outward from S" constraint is removed. It is straightforward to obtain a lower bound on $\min_{\theta \in \partial S} ||v(\theta)||_\infty$ with a minor relaxation to either $\widehat{V}(\gamma)$ or $\widehat{V}(\gamma, u)$: simply remove the $d_{s(e)}^{-1} f_{s(e)} - d_{t(e)}^{-1} f_{t(e)}$ constraints from (1.6f) and (1.6g). This lower bound can be used in place of the left-hand side of (1.11).

In the IEEE 24-bus test case, we use random sampling to identify a point $\gamma \in [\gamma_0, \gamma^*]$ for which $\min_{\theta \in \partial S} ||v(\theta)||_\infty \geq 0.0435$, with respect to the set $S = \Delta(\gamma) \cap \Omega_{u_0}$, $u_0 = \mathbb{0}_{11}$. The arc lengths in this particular γ range from 18.5 to 22.1 degrees, with a median of 20.3 degrees. Therefore, Theorem 1.12 guarantees that the IEEE 24-bus steady-state is robust to any perturbations in parameters for which $||\xi(\theta_0)||_\infty + \max_{\theta \in \partial S} ||\xi(\theta)||_\infty < 0.0435$.

In the remaining subsections, we will apply Theorem 1.12 to particular modes of disturbances: fluctuations in nominal power injections, changes in nodal voltages, and changes in branch admittances.

1.5.2 Perturbations of Nominal Injections

Suppose that the perturbed model is identical to the original model, except the vector of nominal frequency deviations has been shifted to $p^* + \Delta p^*$. It is then clear from (1.5) that the vector of frequency deviations suffers the perturbation $\xi(\theta) = D^{-1}\Delta p^*$. This quantity is constant, and condition (1.11) reduces to the condition

$$||D^{-1}\Delta p^*||_\infty < \frac{1}{2} \min_{\theta \in \partial S} ||v(\theta)||_\infty$$

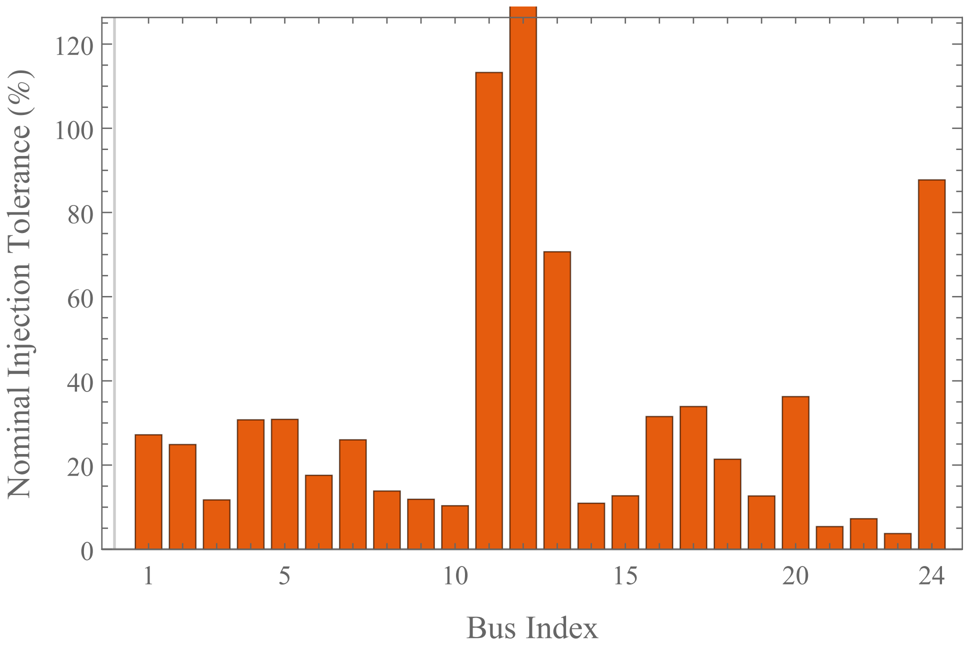

Figure 1.4: Relative tolerances of nominal power injections in the IEEE 24-bus system. Each bar indicates the range of perturbations (as a percentage of the nominal value) with respect to which the transient stability certificate still holds. Note that these perturbations may occur simultaneously. Also note that bus 12 has a tolerance of 1500% due to its small nominal injection.

for some $S = \Delta(\gamma) \cap \Omega_{u_0}$, with $\gamma \in [\gamma_0, \gamma^*]$. In the IEEE 24-bus test case, a sufficient condition is $||\Delta p^*||_\infty < 0.217$ pu. The median bus in this test case has a nominal injection magnitude of 0.94 pu, so our certificate guarantees that the system is robust to disturbances of 23% in the nominal injection of this bus. Figure 1.4 plots the relative tolerance of all buses in the system.

1.5.3 Perturbations of Voltage and Admittance Magnitudes

Next, we consider perturbations to nodal voltage magnitudes and branch admittance magnitudes. Both of these values are encoded in the $\tilde{a}_{ij}$ and a_{ij} parameters, so these perturbations can be represented with perturbations $\Delta\tilde{a}_{ij}$ and Δa_{ij}. The entries of the corresponding perturbation vector are

$$\xi_i(\theta) = d_i^{-1} \sum_{j \in \mathcal{N}(i)} \Delta\tilde{a}_{ij} + \Delta a_{ij} \sin(\theta_i - \theta_j - \phi_{ij})$$

For simplicity, assume that $\Delta\tilde{a}_{ij} \leq 0$ and $\Delta a_{ij} \leq 0$ (i.e., there is a loss in voltage magnitudes or branch admittances). In order to compute $||\xi(\theta_0)||_\infty$, we first compute $\eta_{ij} = \sin(\theta_i - \theta_j - \phi_{ij})$ using θ_0, so that

$$||\xi(\theta_0)||_\infty = \max_i \left\{ d_i^{-1} \left| \sum_{j \in \mathcal{N}(i)} \Delta\tilde{a}_{ij} + \eta_{ij}\Delta a_{ij} \right| \right\}$$

Similarly, defining $\bar{\eta}_{ij} = \max\{\sin(\gamma_{\{i,j\}} - \phi_{ij}),\ \sin(\gamma_{\{i,j\}} + \phi_{ij})\}$ as an upper bound on $|\sin(\theta_i - \theta_j - \phi_{ij})|$ for $\theta \in \Delta(\gamma)$, we can bound

$$\max_{\theta \in \partial S} ||\xi(\theta)||_\infty \leq \max_i \left\{ -d_i^{-1} \left(\sum_{j \in \mathcal{N}(i)} \Delta\tilde{a}_{ij} + \bar{\eta}_{ij}\Delta a_{ij} \right) \right\}$$

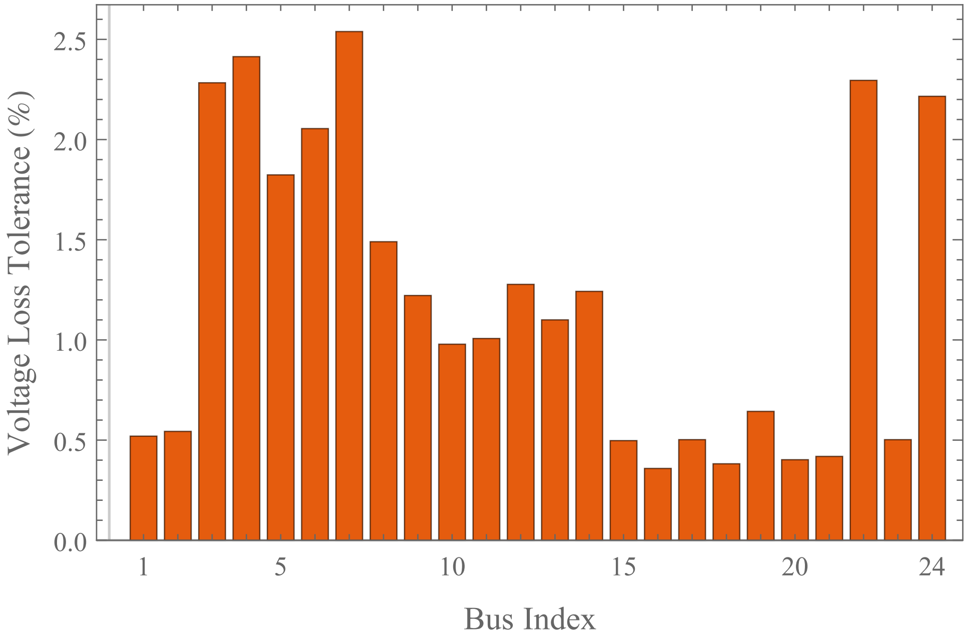

Figure 1.5: Tolerance for voltage loss in the IEEE 24-bus system, as a percentage of the nominal voltage magnitude. If an individual bus loses the fraction of voltage magnitude indicated by its corresponding bar in the chart, then transient stability in the post-fault system is guaranteed.

Then condition (1.11) is satisfied if the sum of these two quantities is less than $\max_{\theta\in\partial S} ||v(\theta)||_\infty$. Note that this condition is a set of linear constraints on $\Delta\tilde{a}_{ij}$ and Δa_{ij}.

For a simple illustrative example, suppose that one particular bus $\ell \in \mathcal{V}$ suffers a loss in voltage magnitude, so that $E_\ell \to (1-\alpha)E_\ell$ for some $\alpha \in (0,1]$. Then $\Delta\tilde{a}_{\ell j} = -\alpha(2-\alpha)\tilde{a}_{\ell j}$, $\Delta a_{\ell j} = -\alpha a_{\ell j}$, and $\Delta a_{j\ell} = -\alpha a_{j\ell}$ for all $j \in \mathcal{N}(\ell)$, while the remaining perturbations are zero. In the IEEE 24-bus test case, we compute the largest value of α that satisfies the previous equation for each $\ell \in \mathcal{V}$, using 0.0435 for the right-hand side of the bound. These largest α are plotted in Figure 1.5. The median bus can tolerate a 1% loss of voltage magnitude before Δa_{ij} and $\Delta\tilde{a}_{ij}$ violate the above bound. This is much more restrictive than the bound for nominal power injections, which is to be expected, given that we used a conservative upper bound on $\max_{\theta\in\partial S} |\xi_i(\theta)|$ instead of the exact value.

1.6 Conclusion

In this chapter, we study transient stability in power networks consisting of droop-controlled inverters and frequency-dependent loads. We extend the notion of transient stability to include not only frequency synchronization but also operating constraints on nodal frequencies, angle differences, power injections, ramping, and storage reserves. To analyze the transients, we introduce a physically-meaningful Lyapunov-like function, and we re-cast the transient stability problem as an optimization problem that admits an efficient relaxation. We show that incorporating information from loop flows (in the form of the winding vector) can make these transient stability certificates less conservative. Finally, we show how these certificates can be used to quantify the size of parameter disturbances to which the system is robust.

The model we use in this chapter is, of course, a highly simplified model for frequency dynamics. Nonetheless, we hope that this work provides a step toward understanding the fundamental behavior of future low-inertia power grids. Extensions of this work may offer rigorous answers to open theoretical questions about these systems. At what scale do the ubiquitous grid-following inverters harm system stability, and how can power engineers use droop-controlled inverters to mitigate this effect? How do legacy high-inertia generators affect transient behavior in power networks that are dominated by inverters? Future research may enrich this work with models of different types of generators, to better-understand frequency dynamics as power grids transition to low-inertia power sources.

1.7 Proofs

Proof of Lemma 1.5. Let θ be the minimizing argument of Problem 1.1, let $y_e = d_{\mathrm{cc}}(\theta_i, \theta_j)$ be the counterclockwise angle difference across each branch $e \in \mathcal{E}$, and define f, η^+, and η^- according to (1.6b)–(1.6d). Because $\theta \in \partial\Delta(\gamma)$, (1.6e) holds because $\theta \in \mathrm{cl}(\Delta(\gamma))$, and there exists some edge $e \in \mathcal{E}$ such that $y_e = \pm\gamma_e$. In the case where $y_e = \gamma_e$, let $z_e^+ = 1$, so the two linear constraints in (1.6f) activate. The first constraint is just $y_e = \gamma_e$, which we have assumed is true. The second constraint, $d_{s(e)}^{-1} f_{s(e)} - d_{t(e)}^{-1} f_{t(e)} \geq 0$, holds because $D^{-1}f(\theta) = D^{-1}f$ points outward from $\Delta(\gamma)$. Thus (1.6f) is satisfied, and (1.6g) and (1.6h) are satisfied by setting all other entries of z^+ and z^- equal to zero. In the case where $y_e = -\gamma_e$, we employ a complementary argument with $z_e^- = 1$. In both cases, all constraints are satisfied, so $(f, y, \eta^+, \eta^-, z^+, z^-)$ is feasible. Finally, the cost function (1.6a) is equal to $V(\theta)$ (the cost function of Problem 1.1) when evaluated at θ. Hence $\widehat{V}(\gamma) \leq V^*(\partial\Delta(\gamma))$.

To see that equality holds in the tree case, let us consider the argmin (f, y, η, z^+, z^-) of Problem 1.2. Because G is a tree and $|y| \leq \gamma$, there always exists $\theta \in \Delta(\gamma)$ for which $y_e = d_{\mathrm{cc}}(\theta_{s(e)}, \theta_{t(e)})$ for all $e \in \mathcal{E}$. Then (1.6e)–(1.6h) ensure that $\theta \in \partial\Delta(\gamma)$, and (1.6a)–(1.6d) ensure that the cost function of Problem 1.2 is identical to $V(\theta)$. Hence $V^*(\partial\Delta(\gamma)) \leq \widehat{V}(\gamma)$ if G is acyclic. □

The proof of Lemma 1.8 proceeds similarly, but it uses some properties of the winding partition of the n-torus. Readers unfamiliar with the winding partition are encouraged to read 1.4.1 and glance at [68]. The property that we need is the following lemma, which shows how the winding partition relates to the boundaries of phase-cohesive sets:

Lemma 1.13. *Let Ω_u be a winding cell, and let $\gamma \in (0, \pi\mathbb{1}_m)$. Consider the set $S = \Delta(\gamma) \cap \Omega_u$. Then $\partial S = \partial\Delta(\gamma) \cap \Omega_u$.*

Proof. We first argue that $\mathrm{cl}(\Delta(\gamma)) \cap \partial\Omega_u = \emptyset$. If $\theta \in \mathrm{cl}(\Delta(\gamma))$, then $|\theta_i - \theta_j| < \pi$ for all $\{i,j\} \in \mathcal{E}$ (by the assumption that $\gamma < \pi \mathbb{1}_m$). Then there is a neighborhood around θ within which the winding numbers around each cycle do not change, so θ belongs to the interior of a winding cell.

First, using elementary properties of topology, we have

$$\partial S \subseteq \mathrm{cl}(S) \subseteq \mathrm{cl}(\Delta(\gamma)) \cap \mathrm{cl}(\Omega_u)$$

Because $\partial S \subseteq \mathrm{cl}(\Delta(\gamma))$, we have that $\partial S \cap \partial\Omega_u = \emptyset$. Therefore, from the elementary property $\partial S \subseteq \partial\Delta(\gamma) \cup \partial\Omega_u$, we obtain the result $\partial S \subseteq \Delta(\gamma)$. Furthermore, because $\partial S \subseteq \mathrm{cl}(\Omega_u)$ but $\partial S \cap \partial\Omega_u = \emptyset$, we have that $\partial S \subseteq \mathrm{int}(\Omega_u) \subseteq \Omega_u$. Hence $\partial S \subseteq \partial\Delta(\gamma) \cap \Omega_u$.

To show equality, we invoke the winding partition to write

$$\begin{aligned}\partial\Delta(\gamma) &= \partial\left(\bigcup_{v \in \mathrm{Img}(w_\Sigma)} \Delta(\gamma) \cap \Omega_v\right) \\ &\subseteq \bigcup_{v \in \mathrm{Img}(w_\Sigma)} \partial(\Delta(\gamma) \cap \Omega_v)\end{aligned}$$

For $v \neq u$, the sets Ω_u and $\partial(\Delta(\gamma) \cap \Omega_v)$ are disjoint, since we have shown that $\partial(\Delta(\gamma) \cap \Omega_v) \subseteq \Omega_v$. Thus, intersecting both sides of the equation with Ω_u, we obtain $\partial\Delta(\gamma) \cap \Omega_u \subseteq \partial(\Delta(\gamma) \cap \Omega_u) = \partial S$. This completes the proof. □

We now prove Lemma 1.8.

Proof of Lemma 1.8. The statement that $\widehat{V}(\gamma) \leq \widehat{V}(\gamma, u)$ is obvious, since Problem 1.2 is a relaxation of Problem 1.3.

To show that $\widehat{V}(\gamma, u) \leq V^*(\partial S)$, let θ be the minimizing argument of Problem 1.1. As in the proof of Lemma 1.5, select the values of the decision variables (f, y, η, z^+, z^-) accordingly to satisfy (1.6b)–(1.6h), thereby ensuring that the cost function (1.6a) is equal to $V(\theta)$. Because $\theta \in \Omega_u$, [68, Theorem 3.5] guarantees the existence of $x \in \mathbb{1}_n^{\mathsf{T}}$ such that $y = B^{\mathsf{T}}x + 2\pi C_\Sigma^\dagger u$. Multiplying across by C_Σ, we obtain

$$C_\Sigma y = C_\Sigma B^{\mathsf{T}} x + 2\pi C_\Sigma C_\Sigma^\dagger u = 2\pi u$$

We have performed two simplifications in this equation. First, $\mathrm{Img}(C_\Sigma^{\mathsf{T}})$ is the cycle space, which is identical to $\ker(B)$ [20, Theorem 9.5], so the $C_\Sigma B^{\mathsf{T}} x$ term vanishes. Second, the summation structure in (1.7) implies that $u \in \mathrm{Img}(C_\Sigma)$, so the orthogonal projection matrix $C_\Sigma C_\Sigma^\dagger$ has no effect on u. Thus (f, y, η) satisfies all of the constraints of Problem 1.3, so $\widehat{V}(\gamma, u) \leq V^*(\partial S)$.

Next, we will show that $V^*(\partial S) \leq \widehat{V}(\gamma, u)$. Let (f, y, η, z^+, z^-) be the minimizing argument of Problem 1.3. Consider the equation $y = B^{\mathsf{T}}x + 2\pi C_\Sigma^\dagger u$. Note that $\mathrm{Img}(B^{\mathsf{T}}) = (\mathrm{Img}(C_\Sigma^\dagger))^\perp$, so we can decompose $y = y_1 + y_2$ with $y_1 \in \mathrm{Img}(B^{\mathsf{T}})$ and $y_2 = \mathrm{Img}(C_\Sigma^\dagger)$, and the equation can be split into $y_1 = B^{\mathsf{T}}x$ and $y_2 = 2\pi C_\Sigma^\dagger u$. The first equation has a unique solution $x \in \mathbb{1}_n^\perp$, while the second equation is true because $C_\Sigma y = 2\pi u$, so there is a unique point $x \in \mathbb{1}_n^\perp$ that satisfies $y = B^{\mathsf{T}}x + 2\pi C_\Sigma^\dagger u$. It follows from [68, Theorem 3.5] that there exists $\theta \in \Omega_u$ such that $y_e = d_{\mathrm{cc}}(\theta_{s(e)}, \theta_{t(e)})$ for all $e \in \mathcal{E}$. From the remaining constraints in Problem 1.3, we can see that $\theta \in \partial\Delta(\gamma) \cap \Omega_u$, so it follows from Lemma 1.13 that $\theta \in \partial S$. Furthermore, the constraints imply that the velocity vector is pointed outward. Thus θ is within the feasible set of Problem 1.1, and the identical values of the cost functions imply that $V^*(\partial S) \leq \widehat{V}(\gamma, u)$. □

Chapter 2

Network Resource Allocation in Epidemic Models

This chapter was first published in *IEEE Transactions on Automatic Control* [117].[1]

The basic reproduction number R_0 is a fundamental quantity in epidemiological modeling, reflecting the typical number of secondary infections that arise from a single infected individual. While R_0 is widely known to scientists, policymakers, and the general public, it has received comparatively little attention in the controls community. This note provides two novel characterizations of R_0: a stability characterization and a geometric program characterization. The geometric program characterization allows us to write R_0-constrained and budget-constrained optimal resource allocation problems as geometric programs, which are easily transformed into convex optimization problems. We apply these programs to allocating vaccines and antidotes in numerical examples, finding that targeting R_0 instead of the spectral abscissa of the Jacobian matrix (a common target in the controls literature) leads to qualitatively different solutions.

[1] ©2022 IEEE. Reprinted, with permission, from Kevin D. Smith and Francesco Bullo, *Convex Optimization of the Basic Reproduction Number*, October 2022.

2.1 Introduction

Perhaps the most important parameter in an epidemic is the basic reproduction number. This number, denoted R_0, is the number of secondary infections that arise from a typical infected individual within an otherwise completely susceptible population. R_0 is a widely-known term, especially since 2020, when articles with "R_0" in the title ran in mainstream publications like *The New York Times* and *The Wall Street Journal*. Since R_0 is an intuitive and widely-known quantity, one might also expect it to appear frequently in the controls literature on epidemics, but this is not the case.

Instead, the literature tends to focus on two other major approaches to epidemic control. First, in the *optimal control framework*, parameters or control inputs are chosen to minimize some cost function integrated along the model trajectory [108, 84, 60, 123]. These trajectories seldom admit closed-form solutions, so this approach generally requires model-specific analysis and numerical solutions of Pontryagin's conditions [108, 84], potentially large-scale optimization to embed discrete-time dynamics [60], or linearization and a discount factor to ensure convergence [123]. The second major approach is the *spectral optimization framework*, in which resources are allocated to minimize the spectral abscissa of the model's Jacobian matrix about some disease-free equilibrium [101, 128, 99, 92, 65]. If the Jacobian is stable, then the abscissa represents the rate at which the trajectory converges to this equilibrium, so minimizing the (negative) abscissa leads to a faster-decaying epidemic. Spectral optimization is based on a linear approximation of the model, but it is nonetheless an appealing framework for resource allocation, since the spectral abscissa can be directly evaluated from model parameters (without computing a trajectory).

The spectral abscissa is closely related to R_0. They are equivalent threshold parameters for whether the epidemic spreads or decays: in compartmental epidemic models (under reasonable assumptions), the epidemic enters an exponential growth phase if and only if the abscissa is positive, if and only if $R_0 > 1$ [37]. Furthermore, intuitively, both quantities reflect the *rate* at which the epidemic spreads or decays. But it is important to note that the abscissa and R_0 are different quantities. In fact, through proper choice of infection and recovery rates in the Kermack-McKendrick SIR model, one can achieve *any* pair of values for the abscissa α and reproduction number R_0 such that $R_0 > 0$ and $\operatorname{sgn}(\alpha) = \operatorname{sgn}(R_0 - 1)$. Thus, while the intuition for these two quantities is similar, minimizing the abscissa will generally lead to a different allocation of resources than minimizing R_0 directly.

To our knowledge, there is no work in the literature that focuses on directly minimizing or constraining R_0 in the resource allocation problem. Motivated by the ubiquity of R_0 in epidemiology and its popularity in the public discourse around COVID-19, this note provides theoretical foundations to fill in this gap.

Contributions We propose a modification of the spectral optimization framework to operate on R_0 instead of on the spectral abscissa. We offer three primary contributions:

(i) We provide two novel characterizations of R_0 in compartmental epidemic models. One characterization relates R_0 to the stability of perturbations to the Jacobian matrix, and the other expresses R_0 as a geometric program, which can be transformed into a convex optimization problem.

(ii) We define two R_0-based optimal resource allocation problems: the R_0-constrained allocation problem, which identifies the lowest-cost allocation to restrict R_0 below a given upper bound; and the budget-constrained allocation problem, which minimizes R_0 with a limited allowance for resource cost. We provide a geometric programming transcription for both of these problems, allowing them to be solved efficiently with off-the-shelf software.

(iii) We present numerical results based on a county-level multi-group SEIR model in California, parameterized using real-world cell phone mobility data. The experiments study the allocation of vaccines and antidotes, a classical problem in spectral optimization. We explain and emphasize the differences between the allocations based on R_0 and the corresponding allocations based on the abscissa.

Organization Section 2.2 introduces the general family of compartmental epidemic models that we consider (§2.2.1), formally defines R_0 (§2.2.2), briefly reviews geometric programming (§2.2.3), and states three key lemmas about Metzler and Hurwitz matrices (§2.2.4). Section 2.3 presents our main theoretical results, including the two new characterizations of R_0 (§2.3.1), and the two R_0-based optimal resource allocation problems and their geometric program transcriptions (§2.3.2). Finally, Section 2.4 presents the numerical experiments.

Notation The matrix $A \in \mathbb{R}^{n \times n}$ is *Metzler* if all its off-diagonal entries are non-negative and is *Hurwitz* if all its eigenvalues have negative real part. Let $\rho(A)$ denote the spectral radius of A. Given $A \in \mathbb{R}^{n \times n}$, let $\operatorname{diag}(A)$ denote the vector in $\mathbb{R}^n$ composed of the diagonal elements of A. Given $x \in \mathbb{R}^n$, let $\operatorname{diag}(x)$ denote the diagonal matrix whose diagonal is x. Thus $\operatorname{diag}(\operatorname{diag}(x)) = x$, and $\operatorname{diag}(\operatorname{diag}(A))$ is a copy of A with all off-diagonal entries set to zero. Given a set S, we write $\operatorname{cl}(S)$ to denote the closure of S.

2.2 Preliminaries

2.2.1 Compartmental Epidemic Models

Compartmental models are a general and widely-used family of epidemic models that divide a population into compartments based on disease state and other demographic factors. This chapter focuses on deterministic epidemic models, in which the number of individuals in each compartment is governed by a system of differential equations. Perhaps the most well-known example is Kermack and McKendrick's SIR model, which has three compartments (susceptible, infected, and recovered), but compartmental models can be arbitrarily complex to capture nuances in the spread of infection between different parts of the population in different disease states. Compartmental models are frequently based on an underlying stochastic model, such that the state variables approximate the expected number of individuals in each compartment.

We consider the general compartmental model in [37], with n infected compartments and m non-infected compartments. The components of this model are as follows. Let $x \in \mathbb{R}^n$ be the expected numbers of individuals in each infected compartment, and let $y \in \mathbb{R}^m$ be the expected numbers of non-infected individuals. The resulting dynamics is

$$\dot{x} = f(x, y) + v(x, y) \tag{2.1a}$$

$$\dot{y} = g(x, y) \tag{2.1b}$$

where f, v, and g are continuously differentiable and defined on non-negative domains. The dynamics of the infected subsystem are decomposed into two vector fields f and v, where f contains the rates at which new infections appear, and v contains rates of transitions that do not correspond to new infections. For example, if infected individuals must pass through a latent disease state before entering an active infectious state (as in the SEIR model), then f captures new infections as they appear in the latent state, while transitions from latent to active infections are contained in v, since the latter are not altogether new infections. This explicit separation of rates corresponding to new infections from all other transitions is crucial to the computation of R_0, and it reflects extra physical interpretation that cannot be inferred from the expression for $\dot{x}$ alone.

Assumption 2.1 (Regularity of f, v, and g)**.** *The vector fields f, v, and g have the following properties:*

(i) $f(x, y) \geq \mathbb{0}_n$ *for all* x *and* y*;*

(ii) $f(\mathbb{0}_n, y) = \mathbb{0}_n$ *and* $v(\mathbb{0}_n, y) = \mathbb{0}_n$ *for all* y*;*

(iii) for all x*,* y*, and* i*,* $x_i = 0$ *implies that* $v_i(x, y) \geq 0$*;*

(iv) for all x*,* y*, and* j*,* $y_j = 0$ *implies that* $g_j(x, y) \geq 0$*.*

Assumption 2.1 collects weak conditions that are obvious from the physical interpretations of f, v, and g. Condition (i) follows from the interpretation of f as a rate at which new infections are created. Condition (ii) ensures that no individuals can transfer into or out of an infected compartment (through new infections or otherwise) if the population is completely free of disease; thus every disease-free state is an equilibrium of (2.1a). Finally, conditions (iii) and (iv) reflect the fact that individuals cannot transition out from an empty compartment.

We also assume that (2.1) admit a disease-free equilibrium point $(\mathbb{0}_n, y^*)$ that is locally asymptotically stable *in the absence of new infections.* That is, if new infections are "switched off" by dropping the vector field f from the dynamics, then the population will return to $(\mathbb{0}_n, y^*)$ even if a small number of infected individuals are introduced.

Assumption 2.2 (Existence of a Stable Equilibrium)**.** *There exists* $y^* \geq \mathbb{0}_m$ *such that* $g(\mathbb{0}_n, y^*) = \mathbb{0}_m$ *and the following Jacobian matrix is Hurwitz:*

$$D \begin{bmatrix} v(\mathbb{0}_n, y^*) \\ g(\mathbb{0}_n, y^*) \end{bmatrix} = \begin{bmatrix} D_x v(\mathbb{0}_n, y^*) & D_y v(\mathbb{0}_n, y^*) \\ D_x g(\mathbb{0}_n, y^*) & D_y g(\mathbb{0}_n, y^*) \end{bmatrix}.$$

The point $(\mathbb{0}_n, y^*)$ satisfying Assumption 2.2 is not necessarily unique, and while it is also an equilibrium point of the full model, it may be unstable when f is no longer ignored.

Under Assumptions 2.1 and 2.2, linearizing the dynamics of (2.1a) about $(\mathbb{0}_n, y^*)$ decouples them from y, and we obtain

$$\dot{x} = (F + V)x \tag{2.2}$$

where $F = D_x f(\mathbb{0}_n, y^*)$ is non-negative and $V = D_x v(\mathbb{0}_n, y^*)$ is Hurwitz and Metzler. We refer the reader to [37, Lemma 1] for the details of this linearization.

2.2.2 Basic Reproduction Numbers

The basic reproduction number is well-known in epidemiology as the typical number of secondary infections that arise from a single infected individual, within an otherwise completely susceptible population. Diekmann, Heesterbeek, and Metz [38] introduced the next generation operator to compute this quantity in general models with structured populations. This approach was later applied by van den Driessche and Watmough [37] specifically to the compartmental model (2.1).

Definition 2.1. *For a compartmental epidemic model* (2.1a)-(2.1b) *satisfying Assumptions 2.1 and 2.2 and with linearization* (2.2) *about* $(\mathbb{0}_n, y^*)$*, the basic reproduction number* *is*

$$R_0 = \rho(FV^{-1}). \tag{2.3}$$

We refer the reader to [38, §2] and [37, §3] for derivations of (2.3) from the epidemiological definition of R_0.

2.2.3 Geometric Programming

Geometric programs are a family of generally non-convex optimization problems that can be transformed into convex optimization problems by a change of variables. Geometric programs enjoy a multitude of applications in engineering and control theory, including the design of optimal positive systems [100], a problem which is closely related to the resource allocation considered in this note. We refer the reader to [16] as a standard introduction to geometric programming and briefly introduce the key concepts in what follows.

A *monomial function* is a map $\mathbb{R}^n_{>0} \to \mathbb{R}_{>0}$ of the form $f(x) = cx_1^{b_1}x_2^{b_2}\cdots x_n^{b_n}$, where $c > 0$ and $b_i \in \mathbb{R}$. A *posynomial function* is a sum of monomial functions. Note that posynomials are closed under addition and multiplication, and that a posynomial divided by a monomial is a posynomial. Given a posynomial function f_0, a set of posynomial functions f_i, $i \in \{1, \dots, m\}$, and a set of monomial functions g_i, $i \in \{1, \dots, p\}$, a geometric program in standard form is:

$$\begin{aligned} \text{minimize}: \quad & f_0(x) \\ \text{variables}: \quad & x > \mathbb{0}_n \\ \text{subject to}: \quad & f_i(x) \leq 1, \; i \in \{1, \dots, m\} \\ & g_i(x) = 1, \; i \in \{1, \dots, p\} \end{aligned}$$

The problem becomes convex after the change of variables $x_i = e^{y_i}$. Off-the-shelf software is available for geometric programs, including the CVX package in MATLAB [50].

2.2.4 Properties of Hurwitz and Metzler Matrices

We now reproduce three lemmas regarding properties of Metzler and Hurwitz matrices that will be necessary for our main results. The first lemma is a standard result characterizing the stability of Metzler matrices (see [22, Theorem 10.14]):

Lemma 2.2 (Metzler Hurwitz Lemma)**.** *Let $M \in \mathbb{R}^{n\times n}$ be a Metzler matrix. The following are equivalent:*

(i) M is Hurwitz,

(ii) M is invertible and $-M^{-1} \geq 0$, and

(iii) there exists $w > \mathbb{0}_n$ such that $Mw < \mathbb{0}_n$.

We borrow the next two results from [37]; the first is a slight restatement of [37, Lemma 5], so we do not include a proof.

Lemma 2.3 (Properties of Hurwitz and Metzler Matrices)**.** *Let $H, M \in \mathbb{R}^{n\times n}$ be Metzler matrices, such that H is Hurwitz and $-MH^{-1}$ is Metzler. The following are equivalent:*

(i) M is Hurwitz, and

(ii) $-MH^{-1}$ is Hurwitz.

The second result is abstracted from the proof of [37, Theorem 2] and we include a self-contained proof.

Lemma 2.4 (Stability of Perturbed Metzler Matrices)**.** *Let $H \in \mathbb{R}^{n\times n}$ be Metzler and Hurwitz, and let $E \in \mathbb{R}^{n\times n}_{\geq 0}$ be a non-negative perturbation matrix. The following are equivalent:*

(i) $H + E$ *is Hurwitz, and*

(ii) $\rho(-EH^{-1}) < 1$.

Proof. Let $A = -(H+E)H^{-1} = -(I_n + EH^{-1})$. Note that A is Metzler, since $-H^{-1} \geq 0$ by Lemma 2.2, so $-EH^{-1} \geq 0$. Then by Lemma 2.3, $H + E$ is Hurwitz if and only if A is Hurwitz. If $\rho(-EH^{-1}) < 1$, then A is clearly Hurwitz. But if $\rho(-EH^{-1}) \geq 1$, then A is not Hurwitz: since $-EH^{-1} \geq 0$, the Perron-Frobenius theorem guarantees that its dominant eigenvalue is real and non-negative, so $-(I_n + EH^{-1})$ has an eigenvalue with non-negative real part. □

2.3 Optimization Framework for R_0

2.3.1 Geometric Program for R_0

The main theoretical result of this chapter is the following theorem, which provides two novel characterizations of R_0:

Theorem 2.5 (Characterizations of R_0)**.** *Consider the linearized epidemic dynamics* (2.2) *with* $F \in \mathbb{R}^{n\times n}_{\geq 0}$ *and* $V \in \mathbb{R}^{n\times n}$ *Hurwitz and Metzler. Write* $V = V_{od} - V_d$*, where* $V_d \geq 0$ *is diagonal and* $V_{od} \geq 0$ *has zero diagonal. The following are characterizations of the basic reproduction number:*

(i) Stability characterization:

$$R_0 = \inf_{r>0} \{ r : F + rV \ \textit{ is Hurwitz} \} \tag{2.4}$$

(ii) Geometric program characterization:

$$R_0 = \inf_{\substack{r>0 \\ w>\mathbb{0}_n}} \left\{ r : \operatorname{diag}(rV_d w)^{-1}(F + rV_{od})w \le \mathbb{1}_n \right\} \tag{2.5}$$

Proof. To prove that (2.4) follows from (2.3), we compute

$$\begin{aligned} \inf_{r>0}\{r : F + rV \text{ is Hurwitz}\} &= \inf_{r>0}\{r : \rho(F(rV)^{-1}) < 1\} \\ &= \inf_{r>0}\{r : \rho(FV^{-1}) < r\} \\ &= \inf_{r>0}\{r : R_0 < r\} = R_0, \end{aligned}$$

where the first step follows from Lemma 2.4. We now use (2.4) to prove (2.5). Let $W = \{w > \mathbb{0}_n : Vw < \mathbb{0}_n\}$ and $\hat{W} = \{w > \mathbb{0}_n : Vw \le \mathbb{0}_n\}$. By Lemma 2.11 (in Appendix 2.6),

$$\begin{aligned} R_0 &= \inf\{r > 0 : F + rV \text{ is Hurwitz}\} \\ &= \inf\{r > 0 : \exists w \in W \text{ s.t. } (F + rV)w < \mathbb{0}_n\} \\ &= \inf\{r > 0 : \exists w \in \hat{W} \text{ s.t. } (F + rV)w \le \mathbb{0}_n\} \\ &= \inf_{r>0,\ w>\mathbb{0}_n} \{r : (F + rV)w \le \mathbb{0}_n\} \end{aligned}$$

In the last step, we note that the $Vw \le \mathbb{0}_n$ constraint is implied by $(F + rV)w \le \mathbb{0}_n$, so we are free to remove it. Manipulating the $(F + rV)w \le \mathbb{0}_n$ constraint into the standard form for geometric programming yields (2.5). □

Remark 2.6 (Degenerate Cases, Pt. I)**.** *The infimum in (2.5) is not always attained. For example, if* $F = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$ *and* $V = -\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$*, then* $R_0 = \inf_{\substack{r>0 \\ w>\mathbb{0}_2}} \left\{ r : r \ge \frac{w_1+w_2}{w_2} \right\} = 1.$ *But there is no feasible point* $w > \mathbb{0}_2$ *that satisfies the inequality constraint with* $r = 1$.

Thus, in general, we cannot replace the infimum in (2.5) with a minimum.

2.3.2 Optimal Resource Allocation

The geometric program characterization (2.5) sets us up to efficiently optimize model parameters to either minimize or constrain R_0. In a manner analogous to [101], we consider two forms of the resource allocation problem: *R_0-constrained allocation*, and *budget-constrained allocation*. In both forms of the resource allocation problem, we suppose that the model parameters F, V_{od}, and V_d depend on a vector of "resources" $\theta \geq \mathbb{0}_k$, and that the cost of a particular allocation of resources is given by a cost function $c(\theta)$. Furthermore, the resources must satisfy some collection of constraints $h(\theta) \leq \mathbb{1}_q$. The dependence on θ must obey the following conditions:

Assumption 2.3 (Resource Dependence)**.** *The resource dependence of the parameters $F(\theta)$, $V_{od}(\theta)$, $V_d(\theta)$, $c(\theta)$, and $h(\theta)$ have the following properties:*

(i) $F(\theta)$, $V_{od}(\theta)$, $c(\theta)$, and $h(\theta)$ are element-wise posynomial functions;

(ii) $V_d(\theta)$ is an element-wise monomial function; and

(iii) the set of feasible allocations $\{\theta \geq \mathbb{0}_k : h(\theta) \leq \mathbb{1}_q\}$ is bounded, and if θ is in this set, then $V_{od}(\theta) - V_d(\theta)$ is Hurwitz.

Conditions (i) and (ii) are necessary to transcribe the allocation problem as a geometric program, while condition (iii) ensures that the matrix parameters F, V_{od}, and V_d satisfy the antecedent of Theorem 2.5 for any feasible allocation. Condition (iii) also ensures the feasible θ are confined to a compact set. Under these assumptions, for all $\theta \in h_{\leq}^{-1}(\mathbb{1}_q)$, the resource dependence of R_0 can be written as

$$R_0(\theta) = \rho\left(F(\theta)(V_{od}(\theta) - V_d(\theta))^{-1}\right). \tag{2.6}$$

Additional resources will typically reduce the rate of new infections or increase the rate at which existing infections are removed. This property is not included in Assumption 2.3, since it is not needed for any of the results in this section. However, if this property is true, then it is useful (albeit unsurprising) to note that $R_0(\theta)$ is weakly decreasing in θ.

Lemma 2.7 (Monotonicity)**.** *Suppose that $F(\theta)$, $V_{od}(\theta)$, and $V_d(\theta)$ satisfy Assumption 2.3. If additionally $F(\theta)$ and $V_{od}(\theta)$ are non-increasing and $V_d(\theta)$ is non-decreasing in θ, then for $\theta, \theta' \in h_{\leq}^{-1}(\mathbb{1}_q)$ with $\theta' \geq \theta$, we have $R_0(\theta') \leq R_0(\theta)$.*

Proof. Let $\theta' \geq \theta$. Since $0 \leq F(\theta') \leq F(\theta)$, $0 \leq V_{od}(\theta') \leq V_{od}(\theta)$, and $V_d(\theta') \geq V_d(\theta) \geq 0$, we can write $F(\theta) = F(\theta') + \Delta F$ and $V(\theta) = V(\theta') + \Delta V(\theta)$ for some matrices $\Delta F, \Delta V \geq 0$. Then

$$\begin{aligned} V^{-1}(\theta) - V^{-1}(\theta') &= (V(\theta') + \Delta V)^{-1} - V^{-1}(\theta') \\ &= -(V(\theta') + \Delta V)^{-1}(\Delta V)V^{-1}(\theta') \\ &\leq 0 \end{aligned}$$

where the last inequality follows from Lemma 2.2, since $V(\theta)$ and $V(\theta')$ are Hurwitz and Metzler, and thus $V^{-1}(\theta) \leq 0$ and $V^{-1}(\theta') \leq 0$. Then

$$\begin{aligned} -F(\theta)V^{-1}(\theta) &= -(F(\theta') + \Delta F)V^{-1}(\theta) \\ &\geq -F(\theta')V^{-1}(\theta) \\ &\geq -F(\theta')V^{-1}(\theta') \end{aligned}$$

Since $-F(\theta)V^{-1}(\theta) \geq 0$ and $-F(\theta')V^{-1}(\theta') \geq 0$, we are guaranteed that

$$\begin{aligned} R_0(\theta) &= \rho(-F(\theta)V^{-1}(\theta)) \\ &\geq \rho(-F(\theta')V^{-1}(\theta')) \\ &= R_0(\theta') \end{aligned}$$

since the spectral radius is weakly increasing in the elements of a non-negative matrix [64, Theorem 8.1.18]. □

We now define the two optimal allocation problems. In the R_0-constrained allocation problem, we identify the cheapest allocation of resources to ensure that $R_0 \leq r_{\max}$, where $r_{\max} > 0$ is some arbitrary threshold. In the budget-constrained allocation problem, some budget $c_{\max} > 0$ is available to spend on resources, and we would like to deploy these limited resources to minimize R_0.

Definition 2.8 (Optimal Allocation Problems). *Let $F(\theta)$, $V_{od}(\theta)$, $V_d(\theta)$, $c(\theta)$, and $h(\theta)$ satisfy Assumption 2.3. We define the following optimization problems:*

(i) Given $r_{\max} > 0$, we say that θ^* is an optimal R_0-constrained allocation if θ^* is a minimizer of

$$\min_{\theta \geq \mathbb{0}_k} \{c(\theta) : h(\theta) \leq \mathbb{1}_q \text{ and } R_0(\theta) \leq r_{\max}\} \tag{2.7}$$

(ii) Given $c_{\max} > 0$, we say that θ^* is an optimal budget-constrained allocation if θ^* is a minimizer of

$$\min_{\theta \geq \mathbb{0}_k} \{R_0(\theta) : h(\theta) \leq \mathbb{1}_q \text{ and } c(\theta) \leq c_{\max}\} \tag{2.8}$$

Assumption 2.3 ensures that $R_0(\theta)$ in (2.6) is well-defined over the feasible sets; furthermore, $R_0(\theta)$ is continuous, since the matrix inverse and spectral radius are continuous functions of the matrix elements. Thus the feasible sets are compact, so the minima of both problems exist.

Using Theorem 2.5, we can construct a pair of geometric programs to solve for optimal R_0-constrained and budget-constrained allocations. For notational convenience, we define a map $p : \mathbb{R}_{>0} \times \mathbb{R}^n_{>0} \times \mathbb{R}^k_{\geq 0} \to \mathbb{R}^n_{>0}$ by

$$p(r, w, \theta) = \operatorname{diag}(rV_d(\theta)w)^{-1}(F(\theta) + rV_{od}(\theta))w \tag{2.9}$$

Under Assumption 2.3, $p(r, w, \theta)$ is posynomial, so the following are geometric programs:

Problem 2.1 (R_0-Constrained Allocation GP)**.** *Given $r_{\max} > 0$ and a tolerance parameter $\tau \geq 0$:*

$$\begin{aligned} \textit{minimize}: &\quad c(\theta) \\ \textit{variables}: &\quad r > 0,\ w > \mathbb{0}_n,\ \theta > \mathbb{0}_k \\ \textit{subject to}: &\quad p(r, w, \theta) \leq \mathbb{1}_n \\ &\quad h(\theta) \leq \mathbb{1}_q \\ &\quad r \leq r_{\max} + \tau \end{aligned}$$

Problem 2.2 (Budget-Constrained Allocation GP)**.** *Given $c_{\max} > 0$:*

$$\begin{aligned} \textit{minimize}: &\quad r \\ \textit{variables}: &\quad r > 0,\ w > \mathbb{0}_n,\ \theta > \mathbb{0}_k \\ \textit{subject to}: &\quad p(r, w, \theta) \leq \mathbb{1}_n \\ &\quad h(\theta) \leq \mathbb{1}_q \\ &\quad c(\theta) \leq c_{\max} \end{aligned}$$

Theorem 2.9 (Geometric Program Transcription)**.** *Let $\theta^* \geq \mathbb{0}_k$, $r_{\max} > 0$, and $c_{\max} > 0$. Let $\mathcal{F}_1(\tau)$ for $\tau > 0$ and $\mathcal{F}_2$ be the sets of feasible points (r, w, θ) for Problems 2.1 and 2.2. The following are true:*

(i) θ^* *is an optimal* R_0*-constrained allocation if and only if the infimum of Problem 2.1 converges to* $c(\theta^*)$ *as* $\tau \to 0_+$ *and there exists* r^*, w^* *such that* $(r^*, w^*, \theta^*) \in \mathrm{cl}(\mathcal{F}_1(\tau))$ *for all* $\tau > 0$.

(ii) θ^* *is an optimal budget-constrained allocation if and only if* $R_0(\theta^*)$ *is the infimum of Problem 2.2 and there exists* r^*, w^* *such that* $(r^*, w^*, \theta^*) \in \mathrm{cl}(\mathcal{F}_2)$.

See Appendix 2.7.1 for the proof.

We note that Problem 2.1 is an arbitrarily accurate approximation of the R_0-constrained allocation problem, controlled by the parameter $\tau \geq 0$. This approximation is necessary due to the closed inequality constraint on R_0 and the representation of R_0 by the infimum in (2.5), which is not always attained:

Remark 2.10 (Degenerate Cases, Pt. II)**.** *In some cases, Problem 2.1 may be infeasible when* $\tau = 0$, *for example, if* $F(\theta) = F$ *and* $V(\theta) = V$ *are the matrices defined in Remark 2.6 and* $r_{\max} = 1$. *Fortunately, the feasible set is nonempty for all* $\tau > 0$, *so we can still consider the limit of solutions to Problem 2.1 as* $\tau \to 0_+$. *This feasibility problem arises due to the constraint on* R_0, *so it is not an issue in Problem 2.2.*

In practice, the issue of an empty feasible set is not of significant concern, since numerical optimization already has inherently limited precision. We suggest solving Problem 2.1 with $\tau = 0$ (and only using a small positive value if the solver reports primal infeasibility).

2.4 Numerical Examples

In the following experiments, we compare R_0-minimizing allocations with abscissa-minimizing allocations. The code used to generate these results is available online.[2]

[2] The MATLAB script and functions used to generate these results is available at `https://www.mathworks.com/matlabcentral/fileexchange/99354-geometric-programs-for-r0`. Running the

2.4.1 Epidemic Model

We adopt a standard multigroup SEIR model (with vital dynamics) for an epidemic in the state of California, where each group corresponds to one of the state's $n = 58$ counties. The SEIR model has two infected states (exposed and infectious) and two non-infected states (susceptible and recovered). Letting $s, e, z, r \in \mathbb{R}^n_{\geq 0}$ denote the expected number of people in each group and disease state, the model dynamics for each group $i \in \{1, \ldots, n\}$ are

$$\begin{aligned} \dot{s}_i &= -\beta_i s_i \sum_{j=1}^{n} a_{ij} z_j & \dot{z}_i &= \gamma_i e_i - \delta_i z_i \\ \dot{e}_i &= \beta_i s_i \sum_{j=1}^{n} a_{ij} z_j - \gamma_i e_i & \dot{r}_i &= \delta_i z_i \end{aligned}$$

It is clear that the model has a disease-free equilibrium $(s_0, \mathbb{0}_n, \mathbb{0}_n, \mathbb{0}_n)$. Linearizing about this point, we obtain

$$\begin{bmatrix} \dot{e} \\ \dot{z} \end{bmatrix} \approx \begin{bmatrix} -\operatorname{diag}(\gamma) & \operatorname{diag}(\beta)\operatorname{diag}(s_0)A \\ \operatorname{diag}(\gamma) & -\operatorname{diag}(\delta) \end{bmatrix} \begin{bmatrix} e \\ z \end{bmatrix}.$$

Because the $\operatorname{diag}(\beta)\operatorname{diag}(s_0)A$ term is the only one corresponding to the creation of new infections, we decompose this Jacobian into the two matrices

$$F = \begin{bmatrix} 0 & \operatorname{diag}(\beta)\operatorname{diag}(s_0)A \\ 0 & 0 \end{bmatrix}, \; V = \begin{bmatrix} -\operatorname{diag}(\gamma) & 0 \\ \operatorname{diag}(\gamma) & -\operatorname{diag}(\delta) \end{bmatrix},$$

where F is non-negative and V is Hurwitz and Metzler.

The model requires a matrix of inter-group contact rates $A \in \mathbb{R}^{n\times n}_{\geq 0}$, which we esti-

code requires an installation of CVX 2.2 and the MOSEK solver.

mated using data from SafeGraph. [3] In particular, we used the Social Distancing Metrics dataset to estimate a matrix $P \in \mathbb{R}^{n \times n}$, where p_{ij} is the daily fraction of people from county i who visited county j, averaged over each day in 2020. Then $(PP^\mathsf{T})_{ij}$ approximates the probability that two random individuals from counties i and j are co-located in the same county on a given day. We set $A = \alpha PP^\mathsf{T}$, where the scalar $\alpha = 2.3667 \times 10^{-7}$ was chosen to ensure $R_0 = 2.5$ when $\beta = 0.1$, $\gamma = 0.2$, and $\delta = 0.1$. Note that α is always multiplied by β, so the only effect of this scalar is to allow us to work with round numbers for β and R_0.

The remaining model parameters are the transmission rates $\beta > \mathbb{0}_n$, incubation rates $\gamma > \mathbb{0}_n$, and recovery rates $\delta > \mathbb{0}_n$ for each group. We used uniform model parameters across each group for simplicity. We generated 2,000 different models by choosing β, γ, and δ for each of 10 (γ and δ) or 20 (β) evenly-spaced values in the range $[0.025, 0.5]$, $[0.05, 0.5]$, and $[0.05, 0.5]$, respectively. The γ and δ range was chosen to allow for a wide range of mean incubation and recovery times (between 2 and 20 days), while the β range was coarsely tuned so that the models have a wide but realistic range of pre-intervention R_0 (95% between 0.23 and 19.38).

2.4.2 Optimal Allocation of Pharmaceuticals

We consider the following optimal resource allocation scenario from [101], in which there are two types of pharmaceutical interventions: vaccines, which reduce the local transmission rates β_i; and antidotes, which increase the local recovery rates δ_i. By allocating vaccines to patch i, we can optimize the local transmission rate within a range

[3] SafeGraph (https://www.safegraph.com) is a data company that aggregates anonymized location data from numerous applications in order to provide insights about physical places, via the SafeGraph Community. To enhance privacy, SafeGraph excludes census block group information if fewer than two devices visited an establishment in a month from a given census block group.

$\beta_i \in [\underline{\beta}_i,\ \overline{\beta}_i]$, where $\overline{\beta}_i \geq \underline{\beta}_i > 0$. The cost of this vaccine allocation is, for all i,

$$f_i(\beta_i) = \frac{\beta_i^{-1} - \overline{\beta}_i^{-1}}{\underline{\beta}_i^{-1} - \overline{\beta}_i^{-1}}. \tag{2.10}$$

Note that the most aggressive allocation has a cost of $f_i(\underline{\beta}_i) = 1$, while allocation of no vaccines at all has a cost $f_i(\overline{\beta}_i) = 0$. The form of (4.7) ensures diminishing returns in the investment of vaccines at each patch. Similarly, by allocating antidotes to patch i, the local recovery rate can be optimized in the range $\delta_i \in [\underline{\delta}_i,\ \overline{\delta}_i]$, with $\overline{\delta}_i \geq \underline{\delta}_i > 0$. The cost of the antidote allocation is, for all i,

$$g_i(\delta_i) = \frac{(\tilde{\delta}_i - \delta_i)^{-1} - (\tilde{\delta}_i - \underline{\delta}_i)^{-1}}{(\tilde{\delta}_i - \overline{\delta}_i)^{-1} - (\tilde{\delta}_i - \underline{\delta}_i)^{-1}}, \tag{2.11}$$

where the parameters $\tilde{\delta}_i > \overline{\delta}_i$ control the shape of the cost curve. The total cost, summing over the local costs of vaccines and antidotes over all patches, is constrained by a budget $c_{\max}$.

In order to perform budget-constrained resource allocation, we must encode the following budget constraint in the standard form for geometric programming:

$$\sum_{i=1}^{n} f_i(\beta_i) + g_i(\delta_i) \leq c_{\max}$$

Since g_i have non-posynomial dependence on δ_i, we replace $1 - \delta_i$ with auxiliary variables η_i, constrained by $\tilde{\delta}_i - \overline{\delta}_i \leq \eta_i \leq \tilde{\delta}_i - \underline{\delta}_i$. Then the posynomial budget constraint is

$$\sum_{i=1}^{n} \frac{\kappa^{-1}\beta_i^{-1}}{\underline{\beta}_i^{-1} - \overline{\beta}_i^{-1}} + \frac{\kappa^{-1}\eta_i^{-1}}{(\tilde{\delta}_i - \overline{\delta}_i)^{-1} - (\tilde{\delta}_i - \underline{\delta}_i)^{-1}} \leq 1 \tag{2.12}$$

where we define a positive constant

$$\kappa = c_{\max} + \sum_{i=1}^{n} \frac{\overline{\beta}_i^{-1}}{\underline{\beta}_i^{-1} - \overline{\beta}_i^{-1}} + \frac{(\tilde{\delta}_i - \underline{\delta}_i)^{-1}}{(\tilde{\delta}_i - \overline{\delta}_i)^{-1} - (\tilde{\delta}_i - \underline{\delta}_i)^{-1}}$$

Altogether, the resource vector is $\theta^\mathsf{T} = \begin{bmatrix} \beta^\mathsf{T} & \eta^\mathsf{T} \end{bmatrix}$, and the constraints are $\underline{\beta}_i \leq \beta_i \leq \overline{\beta}_i$, $\tilde{\delta}_i - \overline{\delta}_i \leq \eta_i \leq \tilde{\delta}_i - \underline{\delta}_i$, and (2.12).

For each experiment, we selected cost parameters based on the pre-intervention SEIR model parameters β and δ. Since pharmaceuticals and vaccines never increase the transmission rate or decrease the recovery rate, we set $\overline{\beta}_i = \beta_i$ and $\underline{\delta}_i = \delta_i$. We chose $\underline{\beta}_i = 0.1\beta_i$ and $\overline{\delta}_i = 2\delta_i$ to reflect a 90% reduction in transmissibility and 50% reduction in mean recovery time at maximum investment, and we selected $\tilde{\delta}_i = 2$ so that $\tilde{\delta}_i > \overline{\delta}_i$.

2.4.3 Results and Discussion

We first set a budget of $c_{\max} = 0.1$ and performed budget-constrained resource allocation to minimize R_0 and the abscissa for each of the 2,000 models. We then simulated the nonlinear post-intervention dynamics for both the R_0-minimized and abscissa-minimized models until convergence.

In 1,270 models, both the R_0-minimized and abscissa-minimized models had $R_0 > 1$, so the number of infected individuals experienced an initial exponential growth phase before peaking and decaying. Figure 2.1 (left) compares the number of active infections at the peak between the R_0-minimized and abscissa-minimized trajectories. In 1,068 (84.1%) of these models, minimizing R_0 led to a smaller peak than minimizing the abscissa. Similarly, Figure 2.1 (right) compares the number of cumulative infections at the end of the simulation. Minimizing R_0 resulted in fewer cumulative cases in 1,056 (83.1%) in the example models. In the remaining models, one or both of the R_0-minimizing or

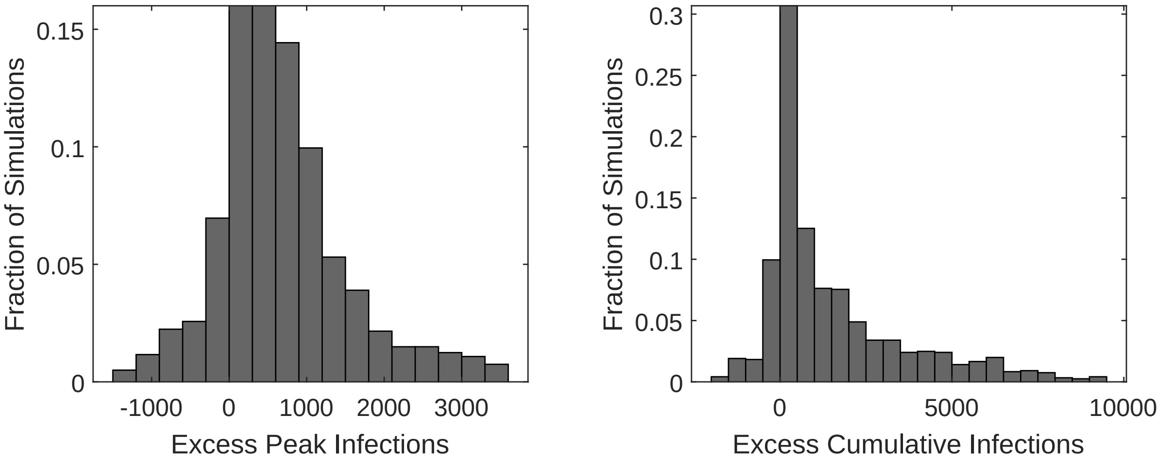

Figure 2.1: Comparison of peak (left) and cumulative (right) infections from minimizing R_0 vs. the abscissa, in models with an initial exponential growth phase. Both histograms show the distribution of how many more infections resulted in the absicssa-minimizing scenario vs. the R_0-minimizing scenario.

abscissa-minimizing allocations led to $R_0 < 1$, so the trajectory immediately decays toward a disease-free equilibrium. It is not meaningful to compare peaks in these models; however, in 96.4% of them, minimizing R_0 resulted in fewer cumulative infections.

Next, we selected three particular models to examine the allocations under various budgets. We chose a low-R_0 model ($\beta = 0.05$, $\gamma = 0.2$, $\delta = 0.2$; $R_0 = 0.625$), a mid-R_0 model ($\beta = 0.1$, $\gamma = 0.2$, $\delta = 0.1$; $R_0 = 2.5$), and a high-R_0 model ($\beta = 0.15$, $\gamma = 0.2$, $\delta = 0.075$; $R_0 = 5.0$), and we repeated the budget-constrained allocations at various budgets. Figure 2.2 (left) plots the cumulative infections for the post-intervention models. Cumulative infections in the R_0-minimized and abscissa-minimized models are very similar at low budgets, but past a budget of 2, minimizing the R_0 leads to a modest decrease in cumulative infections when compared to minimizing the abscissa. (It is not meaningful to plot the peak infections, since $R_0 < 1$ in all post-intervention models with budgets above 2.) Figure 2.2 (right) illustrates a difference in allocation strategies between the two targets, as minimizing R_0 results in a larger share of the budget spent

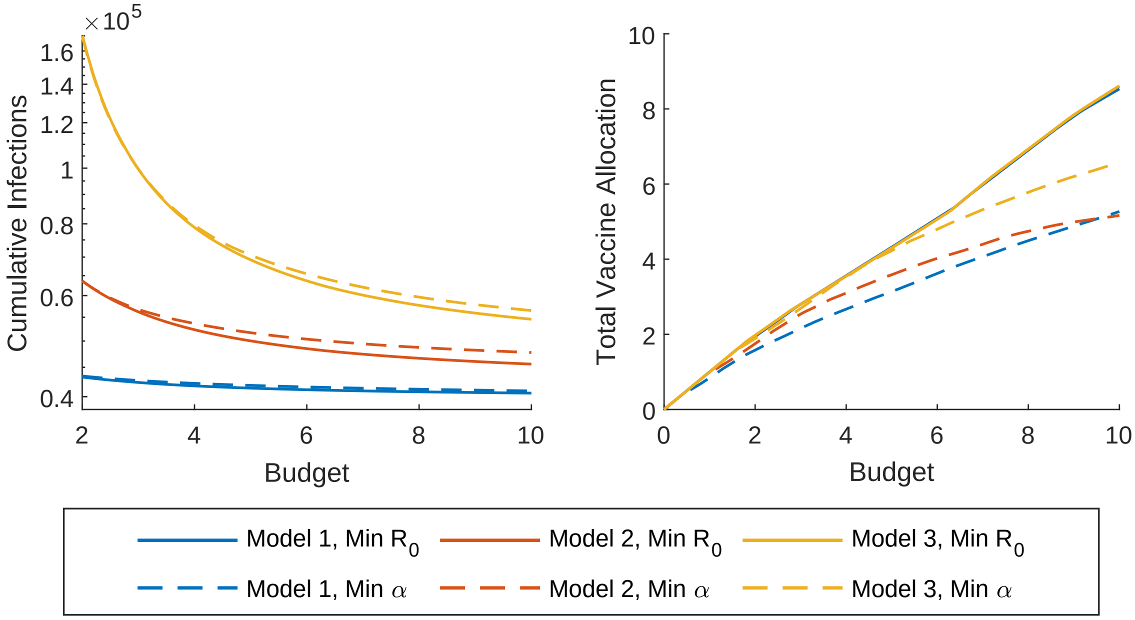

Figure 2.2: Cumulative infections (left) and total budget allocated to vaccines (right) for three models, given various budgets. Solid lines correspond to post-intervention models minimizing R_0, while dashed lines reflect minimizing the abscissa. In these examples, minimizing R_0 results in fewer cumulative infections and a greater fraction of the budget allocated to vaccines.

on vaccines.

2.5 Conclusion

In this note, we have established a new formula for the basic reproduction number of a compartmental epidemic model. We then applied this formula to resource allocation problems that minimize or constrain R_0, transcribing these problems as geometric programs, and we have provided numerical experiments to highlight that targeting R_0 instead of the abscissa can result in qualitatively different solutions. Our results show that R_0 can be a superior target for controlling cumulative and peak infections; however, more work is needed to identify for which models and parameter ranges this is the case. The possible applications of our optimization framework are broad, since it applies to a general class of epidemic models and cost functions. Policymakers should be aware of the limitations of optimal resource allocation: models (and linear models in particular) have limited accuracy, and mathematics does not address the complex social factors of epidemic response. Nonetheless, we believe that this work and its future extensions—coupled with judicious choices of models and cost functions—can provide useful insight for epidemic preparedness and response.

2.6 A Relaxing Lemma

Lemma 2.11 (A Relaxing Lemma). *Let $V \in \mathbb{R}^{n\times n}$ be a Metzler and Hurwitz matrix, and let $F \neq 0$ be a non-negative matrix of the same shape. Let $W = \{w > \mathbb{0}_n : Vw < \mathbb{0}_n\}$, $\hat{W} = \{w > \mathbb{0}_n : Vw \leq \mathbb{0}_n\}$, $R_0 = \inf\{r > 0 : \exists w \in W \text{ s.t. } (F + rV)w < \mathbb{0}_n\}$, and $\hat{R}_0 = \inf\{r > 0 : \exists w \in \hat{W} \text{ s.t. } (F + rV)w \leq \mathbb{0}_n\}$. Then $R_0 = \hat{R}_0$.*

Proof. It is obvious that $\hat{R}_0 \leq R_0$, so we need only show that $\hat{R}_0 \geq R_0$. Before we embark on this task, we will construct useful expressions for $\hat{R}_0$ and R_0. Let I be the (possibly empty) set of indices for which the ith row of F is zero: $F^{(i)} = \mathbb{0}_n$. For any $w \in \hat{W}$, observe that $\{r > 0 : (F + rV)w \leq \mathbb{0}_n\}$ is non-empty if and only if $(Vw)_i = 0$ implies that $i \in I$. Thus, we define

$$\bar{W} = \{w > \mathbb{0}_n : Vw \leq \mathbb{0}_n \text{ and } (Vw)_i < 0 \text{ for all } i \in I^c\}$$

where $W \subset \bar{W} \subset \hat{W}$. Then we can write

$$\begin{aligned}
\hat{R}_0 &= \inf \left(\bigcup_{w \in \hat{W}} \{r > 0 : (F + rV)w \leq \mathbb{0}_n\} \right) \\
&= \inf \left(\bigcup_{w \in \bar{W}} \{r > 0 : (F + rV)w \leq \mathbb{0}_n\} \right) \\
&= \inf_{w \in \bar{W}} \left(\inf\{r > 0 : (F + rV)w \leq \mathbb{0}_n\} \right) = \inf \hat{\mathcal{R}}
\end{aligned}$$

where $\hat{\mathcal{R}} = \{r^*(w) : w \in \bar{W}\}$, and $r^* : \bar{W} \to \mathbb{R}_{\geq 0}$ is the map defined by

$$r^*(w) = \inf\{r > 0 : (F + rV)w \leq \mathbb{0}_n\}, \ \ \forall w \in \bar{W}$$

It is straightforward to solve for $r^*(w)$:

$$r^*(w) = \max_{i \in I^c} \left\{ \frac{(Fw)_i}{|Vw|_i} \right\}, \ \ \forall w \in \bar{W}$$

Similar to $\hat{R}_0$, we have the following expression for R_0:

$$\begin{aligned} R_0 &= \inf\left(\bigcup_{w\in W}\{r>0:(F+rV)w<\mathbb{0}_n\}\right) \\ &= \inf_{w\in W}\left(\inf\{r>0:(F+rV)w<\mathbb{0}_n\}\right) \\ &= \inf_{w\in W}\left(\min\{r>0:(F+rV)w\leq\mathbb{0}_n\}\right) = \inf\mathcal{R} \end{aligned}$$

where $\mathcal{R}=\{r^*(w):w\in W\}$.

The remainder of the proof is to show that R_0 is a lower bound on $\hat{\mathcal{R}}$. Let $\hat{r}\in\hat{\mathcal{R}}$, so that $\hat{r}>0$ and $(F+\hat{r}V)\hat{w}\leq\mathbb{0}_n$ for some $\hat{w}\in\bar{W}$. Let $x>\mathbb{0}_n$ such that $Vx<\mathbb{0}_n$ (which must exist because V is Hurwitz), and for all $t\geq 0$, let $w(t)=\hat{w}+tx$. We can also show that

$$\left|\frac{(Fw(t))_i}{|Vw(t)|_i}-\frac{(F\hat{w})_i}{|V\hat{w}|_i}\right|\leq\kappa_i t,\quad\forall t\geq 0 \text{ and } \forall i\in I^c$$

where

$$\kappa_i=\frac{1}{|V\hat{w}|_i}\left((Fx)_i+\frac{(F\hat{w})_i|Vx|_i}{|V\hat{w}|_i}\right)>0,\quad\forall i\in I^c$$

Then for all $t>0$,

$$\begin{aligned} |r^*(w(t))-r^*(\hat{w})| &= \left|\max_{i\in I^c}\left\{\frac{(Fw(t))_i}{|Vw(t)|_i}\right\}-\max_{j\in I^c}\left\{\frac{(F\hat{w})_j}{|V\hat{w}|_j}\right\}\right| \\ &\leq\left(\max_{i\in I^c}\kappa_i\right)t \end{aligned}$$

Thus, given any $\epsilon>0$, we can choose $t<\epsilon\left(\max_{i\in I^c}\kappa_i\right)^{-1}$ to ensure that $|r^*(w(t))-r^*(\hat{w})|<\epsilon$. Because $w(t)\in W$ for all $t>0$, it is the case that $r^*(w(t))\in\mathcal{R}$ for all $t>0$, so that every open ball around $r^*(\hat{w})$ contains a point in $\mathcal{R}$. Then $r^*(\hat{w})\in\text{cl}(\mathcal{R})$, which implies that $r^*(\hat{w})\geq R_0$. But $r^*(\hat{w})\leq\hat{r}$, and $\hat{r}$ was chosen arbitrarily from $\hat{\mathcal{R}}$, so R_0 is a lower bound on $\hat{\mathcal{R}}$. But $\hat{R}_0$ is the greatest such lower bound, so we conclude that

$\hat{R}_0 \geq R_0$. □

2.7 Proofs

2.7.1 Proof of Theorem 2.9

Let $\mathcal{G}_1, \mathcal{G}_2$ be the sets of feasible points θ for (2.7) and (2.8), respectively. We define a Metzler matrix

$$M(r,\theta) = F(\theta) + rV_{od}(\theta) - rV_d(\theta) \tag{2.13}$$

Since the determinant of $M(r,\theta)$ is a polynomial in r of degree n, for some scalars $a_1, a_2, \ldots, a_n \in \mathbb{C}$, we can write $|M(r,\theta)| = (r-a_1)(r-a_2)\cdots(r-a_n)$. Due to (2.4) in Theorem 2.5, $M(R_0(\theta),\theta)$ must be singular, so $R_0(\theta)$ is a root; then we can assign $a_1, a_2, \ldots, a_\ell = R_0(\theta)$ up to some multiplicity ℓ. Define a "pseudo-determinant" $\mu(r,\theta) = (r-a_{\ell+1})\cdots(r-a_n)$ as the product of the remaining factors, which is real and nonzero for all $r \geq R_0(\theta)$. Then

$$M^{-1}(r,\theta) = \frac{\operatorname{adj}(M(r,\theta))}{(r-R_0(\theta))^\ell \mu(r,\theta)}, \quad \forall r > R_0(\theta)$$

Now, pick $z > \mathbb{0}_n$ arbitrarily, and define

$$w(r,\theta) = -(r-R_0(\theta))^\ell M^{-1}(r,\theta)z, \quad \forall r > R_0(\theta) \tag{2.14}$$

$$w^*(\theta) = \lim_{r \to R_0(\theta^*)^+} w(r,\theta) = -\left(\frac{\operatorname{adj}(M(R_0(\theta),\theta))}{\mu(R_0(\theta),\theta)}\right) z \tag{2.15}$$

For any $r > R_0(\theta)$, (2.4) in Theorem 2.5 implies that $M(r,\theta)$ is Hurwitz, so $-M^{-1}(r,\theta) \geq 0$, and thus $w(r,\theta) > \mathbb{0}_n$. Furthermore, $M(r,\theta)w(r,\theta) < \mathbb{0}_n$, so expanding $M(r,\theta)$ with (4.4) and re-arranging, we obtain $p(r, w(r,\theta), \theta) < \mathbb{1}_n$. We now use $w^*(\theta)$ to formally establish relationships between the feasible sets of both pairs of optimization problems:

Lemma 2.12 (Relating the Feasible Sets)**.** *For each $\tau > 0$, let $\Theta_1(\tau) \subset \mathbb{R}^k$ be the set of θ such that $(r, w, \theta) \in \mathrm{cl}(\mathcal{F}_1(\tau))$ for some r, w. Similarly, let $\Theta_2 \subset \mathbb{R}^k$ be the set of θ such that $(r, w, \theta) \in \mathrm{cl}(\mathcal{F}_2)$ for some r, w. The following are true:*

(i) $\theta \in \mathcal{G}_1 \Longrightarrow (R_0(\theta), w^*(\theta), \theta) \in \mathrm{cl}(\mathcal{F}_1(\tau))$ *for all* $\tau > 0$*,*

(ii) $\theta \in \mathcal{G}_2 \implies (R_0(\theta), w^*(\theta), \theta) \in \mathrm{cl}(\mathcal{F}_2)$*,*

(iii) $\mathcal{G}_1 = \bigcap_{\tau>0} \Theta_1(\delta)$*, and*

(iv) $\mathcal{G}_2 = \Theta_2$*.*

Proof. To prove (i), let $\theta \in \mathcal{G}_1$, so $h(\theta) \leq \mathbb{1}_q$ and $R_0(\theta) \leq r_{\max}$. Fix any $\tau > 0$, and let $\epsilon > 0$. By (2.15), we can choose $r > R_0(\theta)$ such that $||w(r, \theta) - w^*(\theta)|| < \epsilon$ and $|r - R_0(\theta)| < \min\{\tau, \epsilon\}$. Since $p(r, w(r, \theta), \theta) \leq \mathbb{1}_q$ and $r < R_0(\theta) + \tau \leq r_{\max} + \tau$, we have $(r, w(r, \theta), \theta) \in \mathcal{F}_1(\tau)$, so every neighborhood of $(R_0(\theta), w^*(\theta), \theta)$ (by choice of ϵ) contains a point in $\mathcal{F}_1(\tau)$. We prove (ii) by a similar argument (without τ), noting that $\theta \in \mathcal{G}_2$ implies $c(\theta) \leq c_{\max}$.

To prove (iii), we note that (i) implies that $\mathcal{G}_1 \subseteq \bigcap_{\tau>0} \Theta_1(\tau)$. If $\theta \in \Theta_1(\tau)$ for all $\tau > 0$, then $h(\theta) \leq \mathbb{1}_q$ and $R_0(\theta) \leq r_{\max} + \tau$ for all $\tau > 0$, which implies $R_0(\theta) \leq r_{\max}$, and thus $\theta \in \mathcal{G}_1$. Hence $\mathcal{G}_1 \supseteq \bigcap_{\tau>0} \Theta_1(\tau)$ as well. Statement (iv) follows from a similar argument. □

Proof of Theorem 2.9. In order to prove (i), we first define $c^*(\delta)$ as the infimum of Problem 2.1 for all $\tau > 0$, and we define c^* as the minimum cost of the R_0-constrained allocation problem. Noting that $\Theta_1(\tau)$ are nested downward as $\tau \to 0$:

$$c^* = \min \mathcal{G}_1 = \min \bigcap_{\tau>0} \Theta_1(\tau) = \lim_{\tau \to 0^+} \min \Theta_1(\tau) = \lim_{\tau \to 0^+} c^*(\tau)$$

The second step is due to Lemma 2.12, and the third step is a general property of intersections of nested sets. Let θ^* be an optimal R_0-constrained allocation. Then $\theta^* \in \mathcal{G}_1$, so by Lemma 2.12, $(R_0(\theta^*), w^*(\theta^*), \theta^*) \in \mathrm{cl}(\mathcal{F}_1(\tau))$ for all $\tau > 0$, and we have shown that $c^*(\tau) \to c^* = c(\theta^*)$ as $\tau \to 0_+$. On the other hand, if there exist r^*, w^* such that $(r^*, w^*, \theta^*) \in \mathrm{cl}(\mathcal{F}_1(\tau))$ for all $\tau > 0$, then Lemma 2.12 guarantees $\theta^* \in \mathcal{G}$, and $c^*(\tau) \to c(\theta^*)$ implies that $c(\theta^*) = c^*$.

We now prove (ii). Let θ^* be an optimal budget-constrained allocation. Then $\theta^* \in \mathcal{G}_2$, so Lemma 2.12 implies that $(R_0(\theta^*), w^*(\theta^*), \theta^*) \in \mathrm{cl}(\mathcal{F}_2)$. Consider any other point $(r, w, \theta) \in \mathrm{cl}(\mathcal{F}_2)$, and note that Lemma 2.12 also implies $\theta \in \mathcal{G}_2$, so that $R_0(\theta^*) \leq R_0(\theta)$. But $R_0(\theta) \leq r$ by (2.5), so $R_0(\theta^*) \leq r$. Thus $R_0(\theta^*)$ is the min value of r over $\mathrm{cl}(\mathcal{F}_2)$.

Finally, suppose that $(R_0(\theta^*), w^*(\theta^*), \theta^*) \in \mathrm{cl}(\mathcal{F}_2)$ and that $R_0(\theta^*)$ is the infimum of Problem 2.2. Lemma 2.12 guarantees that $\theta^* \in \mathcal{G}_2$. Consider any other point $\theta \in \mathcal{G}_2$, and note that $(R_0(\theta), w^*(\theta), \theta) \in \mathrm{cl}(\mathcal{F}_2)$ as well, so that $R_0(\theta^*) \leq R_0(\theta)$. Therefore θ^* is a minimizer for (2.8), so it is an optimal budget-constrained allocation. □

Chapter 3

Optimal Control of Contracting Systems

This chapter will appear in *IEEE Control Systems Letters* by IEEE [116].[1]

Strongly contracting dynamical systems have numerous properties (e.g., incremental ISS), find widespread applications (e.g., in controls and learning), and their study is receiving increasing attention. This work starts with the simple observation that, given a strongly contracting system, its adjoint dynamical system is also strongly contracting, with the same rate, with respect to the dual norm, under time reversal. As main implication of this dual contractivity, we show that the classic Method of Successive Approximations (MSA), an indirect method in optimal control, is a contraction mapping for short optimization intervals or large contraction rates. Consequently, we establish new convergence conditions for the MSA algorithm, which further imply uniqueness of the optimal control and sufficiency of Pontryagin's minimum principle under additional assumptions.

[1] ©2022 IEEE. Reprinted, with permission, from Kevin D. Smith and Francesco Bullo, *Contractivity of the Method of Successive Approximations for Optimal Control*, December 2022.

3.1 Introduction

Optimal control is generally a difficult problem, and with the exception of some analytically tractable cases, it must be solved numerically. Numerical approaches broadly fall into two categories: direct and indirect methods. Direct methods, like direct collocation and direct shooting methods [126, 103, 10], discretize and approximate the state and/or control to encode the problem as a nonlinear program. Due to their relative simplicity, robustness, and the wide availability of software implementations, direct methods tend to be favored in modern times [10, §4.3], [31].

Indirect methods are an older class of methods based on Pontryagin's minimum principle (PMP), which gives a necessary condition for optimality of a control signal. PMP states that the optimal trajectory must solve a two-point boundary problem, together with a costate, and that the optimal control minimizes a Hamiltonian function at each point in time. Indirect methods search for an input, state trajectory, and costate trajectory that satisfy PMP. Many direct methods, including shooting and collocation, can also be applied as indirect methods to the PMP boundary value problem [74]. Another approach is the Method of Successive Approximations (MSA) [25], also called the Forward-Backward-Sweep algorithm [85], which is the main topic of this chapter.

MSA [75, 80, 4] and its variants [95, 25] are classic approaches that have received renewed attention in the machine learning community [87, 86, 14] as alternatives to gradient descent for training residual neural networks (ResNets). Indeed, a new thrust of machine learning research is to apply control-theoretic techniques to the training of ResNets by viewing these models as forward Euler discretizations of continuous-time control systems [43, 140, 127]. Within this framework, training the ResNet can be viewed as an optimal control problem. As argued in [87, 86], MSA (and its variants) allow for error and convergence analysis and can lead to better training dynamics than gradient descent.

Unfortunately, MSA does not always converge, a problem that is still the subject of ongoing research. In [93], the authors prove convergence criteria based on boundedness and Lipschitz assumptions. Similar bounds are established in [87, 86]. This chapter provides a new set of convergence criteria when MSA is applied to strongly contracting dynamical systems.

The contributions of this chapter are as follows. First, in §3.3, we study the adjoints of nonlinear systems that arise in optimal control theory. We show that adjoints of contracting systems under time reversal are also contracting with the same rate, albeit with respect to the dual norm. This property allows us to prove Grönwall-like and ISS-like bounds on the adjoint dynamics. §3.4 applies these bounds to analyze MSA. Assuming Lipschitz continuity of all relevant maps in the optimal control problem, we obtain a bound on the Lipschitz constant of each MSA iteration. This Lipschitz constant becomes arbitrarily small in the limits of short optimization intervals and large contraction rates, thereby establishing conditions for when the iteration is a contraction mapping. With an additional assumption of pointwise uniqueness of the minimizer of the Hamiltonian, we show that these conditions also lead to uniqueness of the optimal control and sufficiency of PMP.

3.2 Preliminaries

3.2.1 Contracting Dynamics over Normed Vector Spaces

Let $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ be a norm. The dual norm $\|\cdot\|_\star : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is the norm $\|x\|_\star = \sup_{\|y\| \leq 1} y^\top x$. Given a matrix $A \in \mathbb{R}^{n \times n}$, the induced norm of A is $\|A\| = \sup_{\|x\|=1} \|Ax\|$

and the induced logarithmic norm of A is

$$\mu(A) = \lim_{\alpha \to 0^+} \frac{\|I_n + \alpha A\| - 1}{\alpha}.$$

Explicit formulas for the induced (logarithmic) norms are known for the standard $p \in \{1, 2, \infty\}$ norms on $\mathbb{R}^n$ [21, §2.4].

A map $T : X \to Y$ between normed spaces $(X, \|\cdot\|_X)$ and $(Y, \|\cdot\|_Y)$ is Lipschitz continuous if a constant $\ell \geq 0$ exists such that $\|T(x) - T(\bar{x})\|_Y \leq \ell \|x - \bar{x}\|_X$ for all $x, \bar{x} \in X$. The *minimal Lipschitz constant* $\mathrm{Lip}(T)$ is the infimum over ℓ that satisfy this inequality. If T is continuously differentiable, then $\mathrm{Lip}(T) = \sup_{x \in X} \|D_x T(x)\|$, where $D_x T(x)$ denotes the Jacobian matrix of T. Furthermore, if $X = Y = \mathbb{R}^n$, then the *one-sided Lipschitz constant* of T is $\mathrm{osL}(T) = \sup_{x \in X} \mu(D_x T(x))$. A dynamical system $\dot{x} = f(t, x, \dots)$ with a continuously differentiable vector field $f : \mathbb{R}^n \to \mathbb{R}^n$ is said to be *strongly infinitesimally contracting* with rate $c > 0$ if the map $x \mapsto f(t, x, \dots)$ is uniformly one-sided Lipschitz with constant $-c$ for all t and for all inputs.

Strongly contracting systems enjoy numerous properties. As a useful example, we state the following lemma without proof (as it slightly generalizes [21, Theorem 3.15, Corollary 3.16]).

Lemma 3.1 (Grönwall comparison lemma)**.** *Consider a dynamical system*

$$\dot{x}(t) = f(t, x(t), u_1(t), \dots, u_m(t)), \qquad \forall t \geq 0, \tag{3.1}$$

with $x(t) \in \mathbb{R}^n$ and inputs $u_i \in U_i \subseteq \mathbb{R}^{k_i}$ for $i \in \{1, 2, \dots, m\}$. Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$, and let $\|\cdot\|_{U_i}$ be norms on U_i. Assume that

(i) the system (3.1) is strongly infinitesimally contracting with rate $c > 0$, and

(ii) for each $i \in \{1, 2, \dots, m\}$, the maps $u_i \mapsto f(t, x, u_1, \dots, u_i, \dots, u_m)$ are uniformly

Lipschitz continuous with constant ℓ_{f,U_i} for all $t \geq 0$, $x \in \mathbb{R}^n$, and $u_j \in U_j$ with $j \neq i$.

Let $(u_1, \dots, u_m)$ and $(\bar{u}_1, \dots, \bar{u}_m)$ be input signals, and let $x, \bar{x}$ be the corresponding trajectories of (3.1). For all $t \geq 0$,

$$\begin{aligned} \|x(t) - \bar{x}(t)\| &\leq e^{-ct}\|x(0) - \bar{x}(0)\| \\ &+ \sum_{i=1}^{m} \ell_{f,U_i} \int_0^t e^{-c(t-\tau)}\|u_i(\tau) - \bar{u}_i(\tau)\|_{U_i} d\tau. \end{aligned} \tag{3.2}$$

Note that (3.2) still holds when $c \leq 0$, i.e., for expansive systems with a bounded rate of expansion; however, we do not consider such systems in this chapter.

3.2.2 Optimal Control

We study the following optimal control problem:

Problem 3.1 (Optimal control problem)**.** *Consider a dynamical system*

$$\dot{x}(t) = f(t, x(t), u(t)), \quad x(0) = x_0 \in \mathbb{R}^n, \tag{3.3}$$

where f is continuous in all arguments and continuously differentiable in the second and third arguments. Further consider a cost functional

$$J[u] = \int_0^T \phi(t, x(t), u(t))\, dt + \psi(x(T)), \tag{3.4}$$

where $\phi : [0, T] \times \mathbb{R}^n \times U \to \mathbb{R}$ is a running cost *that is differentiable in the second argument, and $\psi : \mathbb{R}^n \to \mathbb{R}$ is a differentiable* terminal cost*. Let $\mathcal{U} = \{u : [0, T] \to U \text{ s.t. } u \text{ measurable}\}$ be a space of permissible control signals, where $T > 0$ and $U \subseteq \mathbb{R}^k$ is a compact set containing $\mathbb{0}_k$. The* optimal control problem *is to find $u^* \in \mathcal{U}$ that minimizes $J[u^*]$.*

An elementary necessary condition for the optimality of a control is Pontryagin's minimum principle (PMP) [5, Theorem 5.10, Theorem 5.11] [17, Theorem 6.3.1, Theorem 6.5.1]:

Theorem 3.2 (Pontryagin's minimum principle)**.** *Let $u^* \in \mathcal{U}$ be an optimal control for Problem 3.1 (if one exists), and let $x : [0,T] \to \mathbb{R}^n$ be the corresponding trajectory of* (3.3)*. There exists a constant $\nu \geq 0$ such that, for all $t \in [0,T]$,*

$$u^*(t) \in \operatorname*{argmin}_{\tilde{u} \in U} H(t, x(t), \lambda(t), \tilde{u}), \tag{3.5}$$

where $H : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \times U \to \mathbb{R}$ is the Hamiltonian

$$H(t, x, \lambda, u) = \lambda^\mathsf{T} f(t, x, u) + \nu \phi(t, x, u) \tag{3.6}$$

and $\lambda : [0,T] \to \mathbb{R}^n$ is the costate trajectory

$$\dot{\lambda}(t) = -D_x f(t, x(t), u(t))^\mathsf{T} \lambda(t) - \nu \phi_x(t, x(t), u(t)) \tag{3.7}$$

with the boundary condition $\lambda(T) = \nu \psi_x(x(T))$.

If $\nu = 0$, the problem is said to be abnormal [5, §6]. Abnormal problems are notoriously difficult, and we do not consider them in this chapter. Barring abnormal problems, we adopt the standard assumption (without loss of generality) that $\nu = 1$.

3.2.3 Method of Successive Approximations

The Method of Successive Approximations (MSA) [25], also called the Forward-Backward Sweep algorithm [85], is a basic approach to computing an input that satisfies PMP. The method iteratively solves the PMP two-point boundary value problem, then updates the control to minimize the new Hamiltonian at each time, as outlined in Algorithm 3.1.

Algorithm 3.1 Method of Successive Approximations

```
Require: initial guess $u^{(0)} \in \mathcal{U}$
for $i = 1, 2, \dots, N$ do
    $x^{(i)} \leftarrow$ trajectory of (3.3) from $x_0$ with input $u^{(i-1)}$
    $\lambda^{(i)} \leftarrow$ trajectory of (3.7) from $\lambda(T) = \psi_x(x^{(i)}(T))$ with inputs $x^{(i)}$ and $u^{(i-1)}$
    $u^{(i)}(t) \leftarrow \mathrm{argmin}_{\tilde{u} \in U} H(t, x^{(i)}(t), \lambda^{(i)}(t), \tilde{u})$ for all $t \in [0, T]$, ties broken arbitrarily
end for
return $u^{(N)}$
```

The algorithm can run for a fixed number of iterations; alternatively, it may terminate when the difference between successive iterates $u^{(i-1)}$, $u^{(i)}$ is within a specified tolerance. Note that each iteration of the algorithm maps a control $u^{(i-1)}$ to a new control $u^{(i)}$, so that each iteration can be thought of as an operator $\mathrm{MSA} : \mathcal{U} \to \mathcal{U}$.

Definition 3.3 (MSA Operator)**.** *Given a control $u \in \mathcal{U}$, let $x : [0, T] \to \mathbb{R}^n$ be the corresponding trajectory of* (3.3)*, and let $\lambda : [0, T] \to \mathbb{R}^n$ be the trajectory of* (3.7) *from $\lambda(T) = \psi_x(x(T))$. Then $\mathrm{MSA}(u)$ is the control that satisfies* (3.5) *with respect to $x(t)$ and $\lambda(t)$ for all $t \in [0, T]$, with ties broken in an arbitrary deterministic manner.*

Definition 3.3 is well-posed if the signal of Hamiltonian-minimizing controls from (3.5) is measurable. When we analyze the MSA algorithm in §3.4, we will impose Lipschitz continuity assumptions that forbid any edge cases where $\mathrm{MSA}(u)$ is not measurable.

3.2.4 Adjoints

Adjoints are familiar from linear systems theory. Given input and output Hilbert spaces $\mathcal{S}_{\text{in}}, \mathcal{S}_{\text{out}}$ and a linear system $G : \mathcal{S}_{\text{in}} \to \mathcal{S}_{\text{out}}$, the adjoint of G is the unique linear system $\tilde{G} : \mathcal{S}_{\text{out}} \to \mathcal{S}_{\text{in}}$ such that $\langle Gu, y \rangle_{\mathcal{S}_{\text{out}}} = \langle u, \tilde{G}y \rangle_{\mathcal{S}_{\text{in}}}$ for all $u \in \mathcal{S}_{\text{in}}$ and $y \in \mathcal{S}_{\text{out}}$. For an LTV system with the usual (A, B, C, D) representation, the adjoint dynamics are

$$\dot{\lambda}(t) = -A(t)^{\mathsf{T}}\lambda(t) - C(t)^{\mathsf{T}}v(t) \tag{3.8a}$$

$$z(t) = B(t)^{\mathsf{T}}\lambda(t) + D(t)^{\mathsf{T}}v(t) \tag{3.8b}$$

with $v \in \mathcal{S}_{\text{out}}$ and $z \in \mathcal{S}_{\text{in}}$ [51, 3.2.4]. The theory of adjoints leads to the duality of controllability and observability and of linear quadratic regulators and estimators [79].

3.3 Contractivity of the Adjoint

This section examines adjoints of nonlinear systems. We first explain how the notion of "adjoint" frequently used in the optimal control literature relates to the adjoint from linear systems. We then prove a simple yet powerful result: that the adjoint of a strongly infinitesimally contracting system is itself strongly infinitesimally contracting, with respect to the dual norm, when integrated backwards in time. This dual contractivity property leads to useful bounds for the evolution of costates, to later be employed in §3.4.

3.3.1 Adjoints of Nonlinear Systems

Nonlinear systems do not properly have adjoints according to the definition in §3.2.4. Instead, the adjoint of the system's linearized variational dynamics is often referred to as its adjoint [75, 32]. Consider the nonlinear system (3.3) with output $y(t) = x(t)$. Let $u(t)$ be an input signal corresponding to a nominal trajectory $x(t)$, let $\tilde{x}(t)$ be the trajectory from $\tilde{u}(t)$. Linearizing the dynamics of $\delta x(t) = \tilde{x}(t) - x(t)$ from $\delta u(t) = \tilde{u}(t) - u(t)$,

$$\begin{aligned}
(\dot{\delta x})(t) &= D_x f(t, x(t), u(t))\delta x(t) + D_u f(t, x(t), u(t))\delta u(t) \\
(\delta y)(t) &= \delta x(t)
\end{aligned}$$

so by (3.8a), the adjoint dynamics are

$$\dot{\lambda}(t) = -D_x f(t, x(t), u(t))^\mathsf{T} \lambda(t) - v(t) \tag{3.9a}$$

$$z(t) = D_u f(t, x(t), u(t))^\mathsf{T} \lambda(t) \tag{3.9b}$$

where $v(t) \in V \subseteq \mathbb{R}^n$. Not coincidentally, the costate dynamics (3.7) from PMP are of the form (3.9a), with a forcing term $v(t) = \phi_x(t, x(t), u(t))$ from the running cost. Indeed, PMP can be derived from the variational linearization described above; see [17, Theorem 2.3.1, Theorem 6.1.1].

3.3.2 Contractivity of the Adjoint

We now examine the adjoints of strongly contracting systems. When the original system is contracting with respect to a norm $\|\cdot\|$, it is natural to study the adjoint system using the dual norm $\|\cdot\|_\star$, as the following lemma suggests.

Lemma 3.4 (Dual Lipschitz constants)**.** *Let $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ be a norm, and let $\|\cdot\|_\star$ be its dual norm. Let $f, g : \mathbb{R}^n \to \mathbb{R}^n$ be a pair of continuously differentiable vector fields, such that $D_x f(x) = D_x g(x)^\mathsf{T}$ for all $x \in \mathbb{R}^n$. Then*

(i) $\mathrm{Lip}_{\|\cdot\|}(f) = \mathrm{Lip}_{\|\cdot\|_\star}(g)$, *and*

(ii) $\mathrm{osL}_{\|\cdot\|}(f) = \mathrm{osL}_{\|\cdot\|_\star}(g)$.

The following is an immediate consequence of Lemma 3.4:

Theorem 3.5 (Dual contraction)**.** *Consider the pair of dynamical systems* (3.3) *and* (3.9a)*. Let $T > 0$ and $c > 0$, let $\|\cdot\|$ be a norm on $\mathbb{R}^n$, and let $\lambda^{\leftarrow}(t) = \lambda(T - t)$ be the time-reversed trajectory of* (3.9a) *(where we study the time-reversed dynamics due to the minus sign in the vector field). The following are equivalent:*

(i) the $x(t)$ system is strongly infinitesimally contracting with respect to $\|\cdot\|$ with rate c, and

(ii) the $\lambda^{\leftarrow}(t)$ system is strongly infinitesimally contracting with respect to $\|\cdot\|_\star$ with rate c.

Proof. Let g be the function

$$g(t, \tilde{\lambda}, \tilde{x}, \tilde{u}) \triangleq \frac{d\lambda^{\leftarrow}(t)}{dt} = D_x f(t, \tilde{x}, \tilde{u})^{\mathsf{T}} \tilde{\lambda} + v(t),$$

For all fixed t, $\tilde{\lambda}$, $\tilde{x}$, and $\tilde{u}$, we have $D_\lambda g(t, \tilde{\lambda}, \tilde{x}, \tilde{u}) = D_x f(t, \tilde{x}, \tilde{u})^{\mathsf{T}}$. Hence, applying Lemma 3.4 to the maps $\tilde{g}(\lambda) = g(t, \lambda, \tilde{x}, \tilde{u})$ and $\tilde{f}(x) = f(t, x, \tilde{u})$, we obtain $\text{osL}_{\|\cdot\|}(\tilde{f}) = \text{osL}_{\|\cdot\|_\star}(\tilde{g})$. Thus the maps $\lambda = g(t, \lambda, \tilde{x}, \tilde{u})$ are uniformly one-sided Lipschitz with constant $-c$ with respect to $\|\cdot\|_\star$, if and only if the maps $x \mapsto f(t, x, \tilde{u})$ have the same property with respect to $\|\cdot\|$. □

3.3.3 Bounds on Adjoint Dynamics

Theorem 3.5 establishes that $\lambda^{\leftarrow}(t)$ is strongly contracting so long as the original system is strongly contracting, so we can exploit standard bounds on contracting systems to bound the evolution of $\lambda(t)$. Before stating these bounds, we impose the following two assumptions:

Assumption 3.1 (Strong contractivity)**.** *The system* (3.3) *is strongly infinitesimally* *contracting with rate $c > 0$, i.e., $\mathrm{osL}(f(t, \tilde{x}, \tilde{u})) \leq -c$ for all $t \in [0, T]$ and $\tilde{u} \in U$. Furthermore, the trajectory of (3.3) on the interval $[0, T]$ with $u(t) = \mathbb{0}_k$ is bounded.*

Assumption 3.2 (Lipschitz continuity, Pt. I)**.** *For all fixed $t \in [0, T]$ and $x \in \mathbb{R}^n$, the map $u \mapsto f(t, \tilde{x}, u)$ from $(\mathbb{R}^k, \|\cdot\|_U)$ into $(\mathbb{R}^n, \|\cdot\|)$ is Lipschitz with constant $\ell_{f,u}$.*

Strong contractivity is a fairly strong assumption. For example, if some $x^* \in \mathbb{R}^n$ is an equilibrium point of the unforced system for all t, strong contractivity implies that x^* is globally exponentially stable (due to Lemma 3.1). Due to Theorem 3.5, the assumption also implies that the adjoint dynamics are also strongly contracting. Consequently, we can prove that all state and costate trajectories remain bounded.

Lemma 3.6 (Boundedness of state and costate)**.** *Consider the system (3.3) and its adjoint (3.7). If the input spaces $U \subset (\mathbb{R}^k, \|\cdot\|_U)$ and $V \subset (\mathbb{R}^n, \|\cdot\|_\star)$ are bounded, then under Assumptions 3.1 and 3.2, there exist bounded sets $X \subset (\mathbb{R}^n, \|\cdot\|)$ and $\Lambda \subset (\mathbb{R}^n, \|\cdot\|_\star)$ such that $x(t) \in X$ and $\lambda(t) \in \Lambda$ for all $t \in [0, T]$ and measurable $u : [0, T] \to U$ and $v : [0, T] \to V$.*

In the remainder of this chapter, we will let $X, \Lambda \subset \mathbb{R}^n$ be the bounded sets guaranteed by Lemma 3.6. In particular, the boundedness of $\lambda(t)$ allows us to impose additional Lipschitz continuity assumptions:

Assumption 3.3 (Lipschitz continuity, Pt. II)**.** *For all fixed $t \in [0, T]$, $\tilde{x} \in X$, $\tilde{u} \in U$, and $\tilde{\lambda} \in \Lambda$,*

(i) the map $x \mapsto D_x f(t, x, \tilde{u})^\mathsf{T} \tilde{\lambda}$ from $(\mathbb{R}^n, \|\cdot\|)$ into $(\mathbb{R}^n, \|\cdot\|_\star)$ is Lipshitz with constant $\ell_{f_x,x}$, and

(ii) the map $u \mapsto D_x f(t, \tilde{x}, u)^\mathsf{T} \tilde{\lambda}$ from $(\mathbb{R}^k, \|\cdot\|_U)$ into $(\mathbb{R}^n, \|\cdot\|_\star)$ is Lipschitz with constant $\ell_{f_x,u}$.

We are now ready to state the first bound on the evolution of the adjoint trajectories.

Theorem 3.7 (Grönwall comparison of costates). *Consider the system* (3.3) *and its adjoint* (3.9a) *with Assumptions 3.1–3.3. Let $u, \bar{u} : [0,T] \to U$ and $v, \bar{v} : [0,T] \to V$ be two pairs of measurable input signals, and let $\lambda, \bar{\lambda} : [0,T] \to \mathbb{R}^n$ be the corresponding adjoint trajectories. Then for all $t \geq 0$,*

$$\begin{aligned} \|\lambda(t) - \bar{\lambda}(t)\|_\star \leq{}& e^{-c(T-t)} \|\lambda(T) - \bar{\lambda}(T)\|_\star \\ &+ \int_t^T e^{-c(\tau - t)} \|v(\tau) - \bar{v}(\tau)\|_\star \, d\tau \\ &+ \ell_{f_x,u} \int_t^T e^{-c(\tau - t)} \|u(\tau) - \bar{u}(\tau)\|_U \, d\tau \\ &+ \frac{\ell_{f_x,x}\ell_{f,u} \sinh(c(T-t))}{c} \int_0^t e^{-c(T-\tau)} \|u(\tau) - \bar{u}(\tau)\|_U \, d\tau \\ &+ \frac{\ell_{f_x,x}\ell_{f,u} e^{-c(T-t)}}{c} \int_t^T \sinh(c(T-\tau)) \|u(\tau) - \bar{u}(\tau)\|_U \, d\tau. \end{aligned} \tag{3.10}$$

Theorem 3.7 provides a somewhat unwieldy bound. We can sacrifice its sharpness to obtain a much simpler incremental ISS property.

Corollary 3.8 (Incremental ISS of adjoint systems). *Under the same hypotheses as Theorem 3.7,*

$$\begin{aligned} \sup_{t \in [0,T]} \|\lambda(t) - \bar{\lambda}(t)\|_\star \leq{}& \|\lambda(T) - \bar{\lambda}(T)\|_\star \\ &+ \kappa \sup_{t \in [0,T]} \|v(t) - \bar{v}(t)\|_\star \\ &+ \left(\ell_{f_x,u}\kappa + \ell_{f_x,x}\ell_{f,u}\kappa^2\right) \sup_{t \in [0,T]} \|u(t) - \bar{u}(t)\|_U. \end{aligned} \tag{3.11}$$

where

$$\kappa = c^{-1}(1 - e^{-cT}). \tag{3.12}$$

3.4 Applications to Optimal Control

Here we show how the contractivity of the adjoint system leads to the contractivity of the MSA iteration, under additional Lipschitz continuity assumptions.

Assumption 3.4 (Lipschitz continuity of cost gradients)**.** *For all fixed $t \in [0, T]$, $\tilde{x} \in X$, and $\tilde{u} \in U$,*

(i) the map $x \mapsto \phi_x(t, x, \tilde{u})$ from $(\mathbb{R}^n, \|\cdot\|)$ into $(\mathbb{R}^n, \|\cdot\|_\star)$ is Lipschitz with constant $\ell_{\phi_x,x}$,

(ii) the map $u \mapsto \phi_x(t, \tilde{x}, u)$ from $(\mathbb{R}^k, \|\cdot\|_U)$ into $(\mathbb{R}^n, \|\cdot\|_\star)$ is Lipschitz with constant $\ell_{\phi_x,u}$, and

(iii) the map $x \mapsto \psi_x(x)$ from $(\mathbb{R}^n, \|\cdot\|)$ into $(\mathbb{R}^n, \|\cdot\|_\star)$ is Lipschitz with constant $\ell_{\psi_x,x}$.

Assumption 3.5 (Lipschitz continuity of the optimum)**.** *There exists a continuous map $h : [0, T] \times X \times \Lambda \to U$ such that*

$$h(t, x, \lambda) \in \operatorname*{argmin}_{u \in U} H(t, x, \lambda, u) \tag{3.13}$$

for all $t \in [0, T]$, $x \in X$, and $\lambda \in \Lambda$, with ties broken in an identical manner as the MSA operator, where for all fixed $t \in [0, T]$, $\tilde{x} \in X$, and $\tilde{\lambda} \in \Lambda$,

(i) the map $x \mapsto h(t, x, \tilde{\lambda})$ from $(\mathbb{R}^n, \|\cdot\|)$ into $(\mathbb{R}^k, \|\cdot\|_U)$ is Lipschitz with constant $\ell_{h,x}$, and

(ii) the map $\lambda \mapsto h(t, \tilde{x}, \lambda)$ from $(\mathbb{R}^n, \|\cdot\|_\star)$ into $(\mathbb{R}^k, \|\cdot\|_U)$ is Lipschitz with constant $\ell_{h,\lambda}$.

Notice that Assumption 3.5 implies that $\mathrm{MSA}(u)$ is measurable for any $u \in \mathcal{U}$. With these Lipshitz assumptions, we can finally bound the Lipschitz constant of the MSA operator.

Theorem 3.9 (Contractivity of MSA)**.** *Suppose that Problem 3.1 is nonsingular and satisfies Assumptions 3.1–3.5, and consider the norm $\|\cdot\|_{\mathcal{U}} : \mathcal{U} \to \mathbb{R}_{\geq 0}$ given by*

$$\|u\|_{\mathcal{U}} = \sup_{t \in [0,T]} \|u(t)\|_U. \tag{3.14}$$

The following are true:

(i) The Lipschitz constant of an MSA iteration with respect to the $\|\cdot\|_{\mathcal{U}}$ norm is bounded by

$$\mathrm{Lip}(\mathrm{MSA}) \leq b_1 \kappa + b_2 \kappa^2 \tag{3.15}$$

where

$$b_1 = \ell_{h,x} \ell_{f,u} + \ell_{h,\lambda} \left(\ell_{\psi_x,x} \ell_{f,u} + \ell_{\phi_x,u} + \ell_{f_x,u} \right) \tag{3.16a}$$

$$b_2 = \ell_{h,\lambda} \ell_{f,u} \left(\ell_{\phi_x,x} + \ell_{f_x,x} \right) \tag{3.16b}$$

(ii) If $b_1 \kappa + b_2 \kappa^2 < 1$, then the MSA *operator is a contraction; hence it has a unique fixed point $\hat{u} \in \mathcal{U}$, the* MSA *iterates $u^{(i)} = \mathrm{MSA}^i(u^{(0)})$ converge to $\hat{u}$ from any initial guess $u^{(0)} \in \mathcal{U}$, and*

$$\|u^{(i)}(t) - \hat{u}(t)\|_U \leq \left(\frac{(b_1 \kappa + b_2 \kappa^2)^i}{1 - b_1 \kappa - b_2 \kappa^2} \right) \|u^{(1)} - u^{(0)}\|_{\mathcal{U}}$$

for all $t \in [0, T]$.

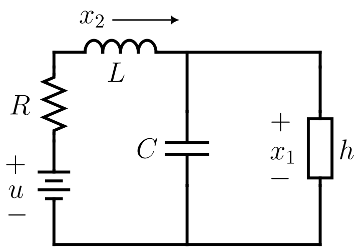

Figure 3.1: Diagram of the nonlinear circuit studied in Section 3.5.

Corollary 3.10 (Uniqueness and Sufficiency)**.** *Under the same hypotheses as Theorem 3.9, if additionally*

(i) the Hamiltonian has a unique minimizer for all $t \in [0, T]$, $x \in X$, and $\lambda \in \Lambda$,

(ii) an optimal control u^* exists, and

(iii) the time horizon T is sufficiently small or the contraction rate c is sufficiently large that $\mathrm{Lip}(\mathrm{MSA}) < 1$*,*

then u^* is the unique optimal control, and PMP is a sufficient condition for optimality.

3.5 Example

For an illustrative example the results, consider the circuit from [76, §1.2.2], depicted in Figure 3.1. The circuit contains a nonlinear resistive element, with a current-voltage relationship $i_h = r(v_h)$ for some twice-differentiable function $r : \mathbb{R} \to \mathbb{R}$ with $r(0) = 0$. We assume that $R > 1$, that $r'(x_1) \geq 1 + \epsilon$ for some $\epsilon > 0$, and that $r''(x_1)$ is bounded for all $x_1 \in \mathbb{R}$. (This assumption allows us to use the $\mathcal{L}_\infty$ norm for simplified analysis; weighted norms can be used to generalize the parameter ranges.) The state variables are $x_1 \in \mathbb{R}$ (voltage across the nonlinear element) and $x_2 \in \mathbb{R}$ (current through the inductor), and the control input $u \in \mathbb{R}$ is the voltage across the source. The dynamics are

$$\dot{x}_1 = \frac{1}{C}\left(-r(x_1) + x_2\right), \quad \dot{x}_2 = \frac{1}{L}\left(-x_1 - Rx_2 + u\right)$$

from an initial condition $x(0) = \mathbb{0}_2$. Our objective is to minimize the cost

$$J[u] = \int_0^T \underbrace{\frac{1}{2}u^2(t)}_{\phi}\, dt + \underbrace{\frac{\gamma}{2}\|x(T) - x^*\|_2^2}_{\psi}$$

for some terminal cost weight $\gamma > 0$, where $x^* \in \mathbb{R}^2$ is an arbitrary target state, and the space of permissible controls is $U = [-u_{\max}, u_{\max}]$ for some $u_{\max} > 0$. Note that $\mathbb{0}_2$ is an equilibrium point of the unforced dynamics.

3.5.1 Examining the Assumptions

This optimal control problem satisfies Assumptions 3.1–3.5, as we demonstrate in the following paragraphs.

Assumption 3.1 The dynamics are strongly infinitesimally contracting with respect to the $\mathcal{L}_\infty$ norm:

$$\begin{aligned}\operatorname{osL}(f) &= \sup_{x \in \mathbb{R}^2} \mu_\infty(D_x f(x)) \\ &= \sup_{x_1 \in \mathbb{R}} \max\left\{\frac{1 - r'(x_1)}{C}, \frac{1 - R}{L}\right\} \\ &= \max\left\{\frac{1 - d_{\min}}{C}, \frac{1 - R}{L}\right\} \triangleq -c < 0\end{aligned}$$

where $d_{\min} = \inf_{x_1 \in \mathbb{R}} r'(x_1) > 1$. Thus Assumption 3.1 is satisfied with contraction rate c.

Assumption 3.2 Given two inputs $u, \bar{u} \in \mathbb{R}$, $\|f(x,u) - f(x,\bar{u})\|_\infty = L^{-1}|u - \bar{u}|$ for all $x \in \mathbb{R}^2$, so Assumption 3.2 is satisfied with $\ell_{f,u} = L^{-1}$.

Reachability Analysis Before we examine Assumption 3.3, it is useful to bound the set of states that are reachable within time T. Lemma 3.1 allows us to compare $x(t)$ with the trajectory at the origin corresponding to zero input:

$$\|x(t)\|_\infty \leq L^{-1} \int_0^t e^{-c(t-\tau)}|u(\tau)|\, d\tau \leq \frac{u_{\max}(1 - e^{-ct})}{cL}$$

In particular, $x(T)$ belongs to a $\mathcal{L}_\infty$ ball centered about the origin, with radius $u_{\max}L^{-1}\kappa$, where κ is defined in (3.12).

Assumption 3.3 Since the Jacobian matrix $D_x f(x,u)$ has no dependence on u, we have $\ell_{f_x,u} = 0$. To evaluate $\ell_{f_x,x}$, note that

$$D_x f(x,u)^\top \lambda = \begin{bmatrix} -C^{-1}r'(x_1) & -L^{-1} \\ C^{-1} & -RL^{-1} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}$$

Then for any $x, \bar{x} \in \mathbb{R}^2$,

$$\begin{aligned} \|D_x f(x,u)^\top \lambda - D_x f(\bar{x},u)^\top \lambda\|_1 &= C^{-1}|\lambda_1||r'(x_1) - r'(x_2)| \\ &\leq C^{-1}\eta|\lambda_1||x_1 - x_2| \end{aligned}$$

where we define $\eta = \sup_{x\in\mathbb{R}} |r''(x)|$. (Note we evaluate the $\mathcal{L}_1$ norm, which is dual to the $\mathcal{L}_\infty$ norm of the state space.) To bound $|\lambda_1|$, we note that Theorem 3.5 implies that the time-reversed costate dynamics $\lambda^{\leftarrow}(t)$ are strongly infinitesimally contracting with rate *c*. Furthermore, the origin is a trajectory, so by Lemma 3.1,

$$\|\lambda^{\leftarrow}(t)\|_1 \leq e^{-ct}\|\lambda^{\leftarrow}(0)\|_1 \leq \|\lambda(T)\|_1, \ \forall t \in [0, T].$$

Since $\lambda(T) = \psi_x(x(T))$, we can then bound

$$\|\lambda(t)\|_1 \leq \|\psi_x(x(T))\|_1 = \gamma\|x(T) - x^*\|_1, \ \forall t \in [0, T].$$

Then for all $t \in [0, T]$,

$$|\lambda_1(t)| \leq \|\lambda(t)\|_1 \leq \gamma\left(\|x^*\|_1 + 2u_{\max}L^{-1}\kappa\right),$$

using the property that $\|x(T)\|_\infty \leq u_{\max}L^{-1}\kappa$. Thus, Assumption 3.3 is satisfied with

$$\ell_{f_x,x} = \frac{\gamma\eta}{C}\left(\|x^*\|_1 + 2u_{\max}L^{-1}\kappa\right).$$

Assumption 3.4 Since the running cost $\phi(u) = u^2$ has no dependence on x, we have $\ell_{\phi_x,x} = 0$ and $\ell_{\phi_x,u} = 0$. Furthermore, for any $x, \bar{x} \in \mathbb{R}^2$, $\|\psi_x(x) - \psi_x(\bar{x})\|_1 = \gamma\|x - \bar{x}\|_1$, so Assumption 3.4 is satisfied with $\ell_{\psi_x,x} = \gamma$.

Assumption 3.5 The Hamiltonian can be written

$$H(x, \lambda, u) = \frac{1}{2}u^2 + \frac{\lambda_2}{L}u + b(x, \lambda)$$

for a constant offset $b(x, \lambda)$. Minimizing the Hamiltonian over $u \in [-u_{\max}, u_{\max}]$ leads to the unique minimizer

$$h(\lambda) = \begin{cases} -u_{\max}, & L^{-1}\lambda_2 > u_{\max} \\ -L^{-1}\lambda_2, & L^{-1}|\lambda_2| \leq u_{\max} \\ u_{\max}, & L^{-1}\lambda_2 < -u_{\max} \end{cases}$$

The map h is Lipschitz in λ with no dependence on x, so Assumption 3.5 is satisfied with $\ell_{h,x} = 0$ and $\ell_{h,\lambda} = L^{-1}$.

3.5.2 Convergence of MSA

Having demonstrated that the optimal control problem satisfies Assumptions 3.1–3.5, we can state the guarantees of Theorem 3.9. Substituting in the Lipschitz constants from the previous section into (3.16a)–(3.16b), we obtain

$$b_1 = \frac{\gamma}{L^2}, \quad b_2 = \frac{\gamma\eta}{CL^2}\left(\|x^*\|_1 + 2u_{\max}L^{-1}\kappa\right)$$

By Theorem 3.9, convergence is guaranteed when

$$\kappa + \frac{\eta\|x^*\|_1}{C}\kappa^2 + \frac{2\eta u_{\max}}{LC}\kappa^3 < \frac{L^2}{\gamma}$$

3.5.3 Numerical Results

Consider a nonlinearity of the form

$$r(v) = \alpha v + \beta\left(\frac{1}{1 - e^{-kv}} - \frac{1}{2}\right) \tag{3.17}$$

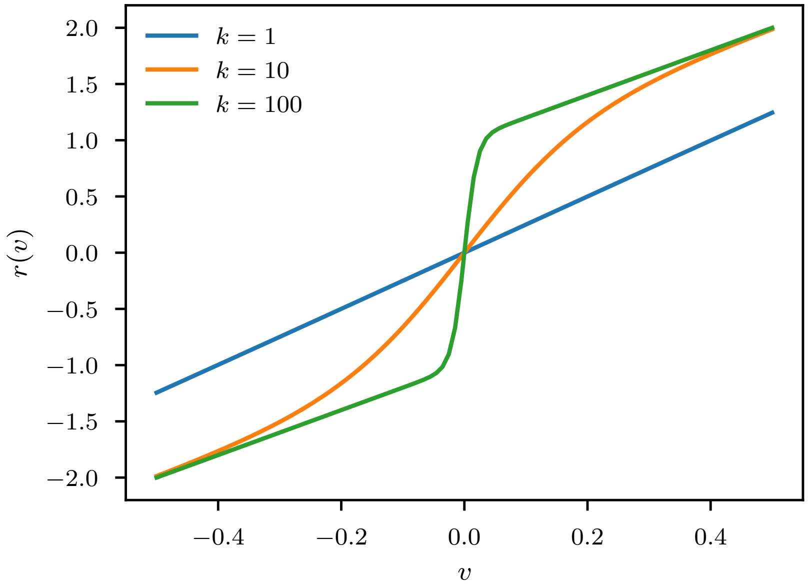

Figure 3.2: Shape of the $r(v)$ function from (3.17), with $\alpha = \beta = 2$, for various values of k.

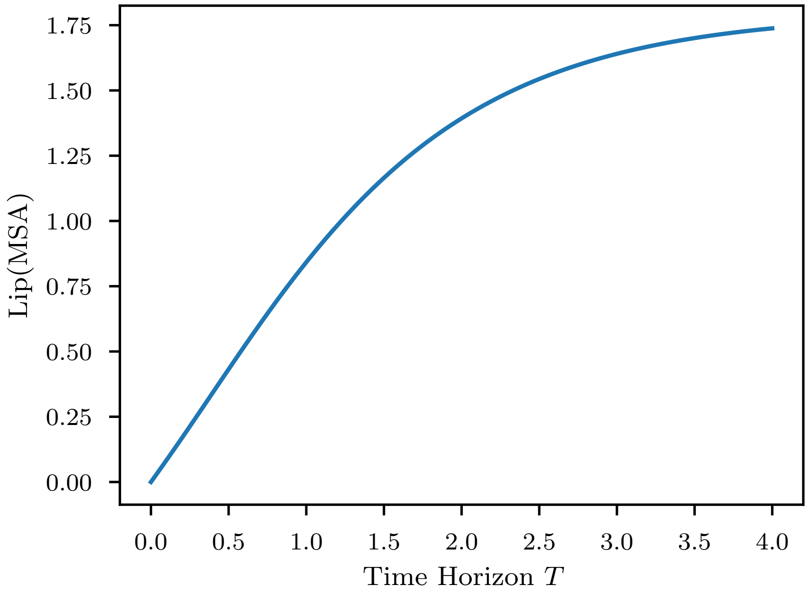

Figure 3.3: Bounds on the Lipschitz constant of the MSA operator at various time horizons, via Theorem 3.9.

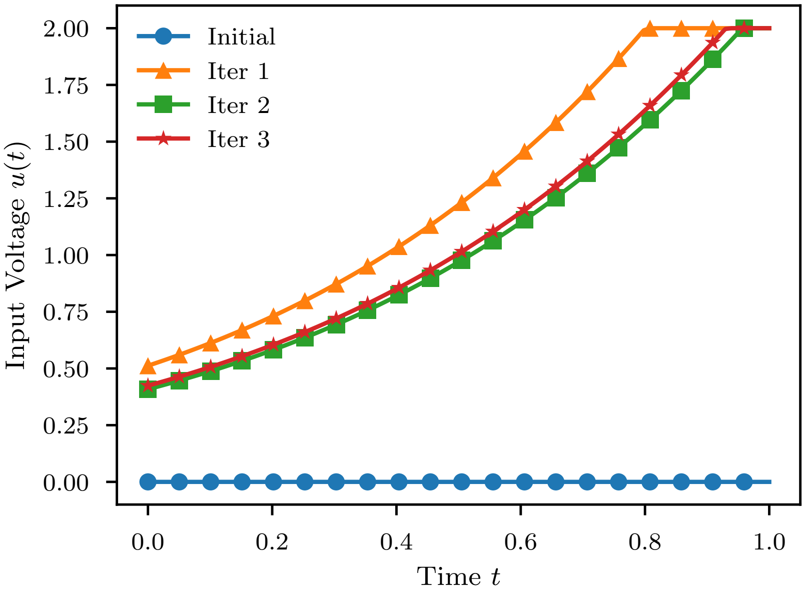

Figure 3.4: Three successive iterates of the MSA algorithm when $\mathrm{Lip}(\mathrm{MSA}) \leq 0.85$, starting from an initial guess $u^{(0)}(t) = 0$.

so that $d_{\min} \triangleq \inf_{v \in \mathbb{R}} r'(v) = \alpha$ and $\eta \triangleq \sup_{v \in \mathbb{R}} |r''(v)| = \beta k^2/(6\sqrt{3})$. Figure 3.2 illustrates this function for various values of the shape parameter k. We select $\alpha = \beta = 2$ and $k = 4$. Furthermore, we select a target state $x^* = (0.1, r(0.1))$, with $u_{\max} = 2$ and terminal cost weight $\gamma = 100$, with model parameters $R = 20$, $L = 11$, and $C = 1$. With these parameters, the contraction rate is $c = 1$, and the upper bound on $\mathrm{Lip}(\mathrm{MSA})$ from Theorem 3.9 is plotted in Figure 3.3. We select a time horizon of $T = 1$, where the bound $\mathrm{Lip}(\mathrm{MSA}) \leq 0.85$ is guaranteed.

In order to implement the MSA algorithm, we use `solve_ivp` from the SciPy package to integrate the state and costate dynamics. This function implements the explicit Runge-Kutta method RK45, and it approximates the solution as a continuous function using quartic interpolation. Starting with an initial guess of $u^{(0)} = 0$, the MSA algorithm quickly converges, with the iterates $u^{(i)}$ for the first three iterations $i \in \{1, 2, 3\}$ depicted in Figure 3.4. Note the rapid decay of the $\mathcal{L}_\infty$ distance between each successive iterate.

3.6 Conclusion

In this chapter, we have examined an indirect method for the optimal control of strongly contracting systems. We have observed that the time-reversed adjoints of such systems are also contracting with the same rate, with respect to the dual norm, leading to useful bounds on the costate trajectory from PMP. Based on this observation, we bounded the Lipschitz constant of each iteration of MSA, demonstrating that the iteration is actually a contraction mapping for sufficiently strongly contractive systems or for sufficiently short horizons. In these cases, MSA is guaranteed to converge to a unique control that satisfies PMP. With an additional assumption on pointwise uniqueness of the minimizer of the Hamiltonian, we showed that this control is indeed the unique optimal control.

The main approach of this chapter, namely using ISS properties of the adjoint to bound the Lipschitz constant of a $\mathcal{U} \to \mathcal{U}$ operator, is quite general and could be applied to many other indirect methods in optimal control. Several variants of MSA, both older [95, 25] and newer [87, 86], could be studied with this type of analysis in future work, possibly with more general convergence criteria. Another practical future direction would be the study of discretized implementation of the forward and backward integration steps, as in [93]. Of course, one could also analyze indirect methods for extensions of the optimal control problem, such as constraints on the terminal state or in the infinite time horizon. An additional interesting direction would be the application of convergence guarantees to model predictive control of contractive nonlinear systems.

3.7 Proofs

3.7.1 Proof of Lemma 3.4

For a general matrix $A \in \mathbb{R}^{n\times n}$, from the definitions of induced norms and dual norms we have

$$\|A\|_\star = \sup_{\|y\|_\star=1} \sup_{\|z\|\leq 1} z^\mathsf{T} A y$$

Swapping the order of the suprema and applying the definition of the dual norm once again yields

$$\|A\|_\star = \sup_{\|z\|\leq 1} \sup_{\|y\|_\star=1} y^\mathsf{T} A^\mathsf{T} z = \sup_{\|z\|\leq 1} \|A^\mathsf{T} z\|_{\star\star}$$

But $\mathbb{R}^n$ with any norm is a reflexive Banach space, so $\|\cdot\|_{\star\star} = \|\cdot\|$, and thus $\|A\|_\star = \|A^\mathsf{T}\|$. We use this fact to prove both statements. Because g is continuously differentiable and $D_x g(x) = D_x f(x)^\mathsf{T}$,

$$\mathrm{Lip}_{\|\cdot\|_\star}(g) = \sup_{x\in X} \|D_x g(x)\|_\star = \sup_{x\in X} \|D_x f(x)\| = \mathrm{Lip}_{\|\cdot\|}(f)$$

and

$$\begin{aligned}\mathrm{osL}_{\|\cdot\|_\star}(g) &= \sup_{x\in X} \lim_{\alpha\to 0^+} \frac{\|I_n + \alpha D_x g(x)\|_\star - 1}{\alpha} \\ &= \sup_{x\in X} \lim_{\alpha\to 0^+} \frac{\|I_n + \alpha D_x f(x)\| - 1}{\alpha} = \mathrm{osL}_{\|\cdot\|}(f)\end{aligned}$$

3.7.2 Proof of Lemma 3.6

Let $\bar{x}(t)$ be the trajectory of (3.3) corresponding to input $\bar{u}(t) = \mathbb{0}_k$. Since (3.3) is strongly infinitesimally contracting, we can use Lemma 3.1 to compare a trajectory $x(t)$

with $\bar{x}(t)$:

$$\|x(t) - \bar{x}(t)\| \leq \frac{\ell_{f,u}}{c}(1 - e^{-cT}) \sup_{\tau \in [0,T]} \|u(\tau)\|_U$$

for all $t \in [0, T]$. Since U and $\bar{x}(t)$ are bounded, $x(t)$ is bounded as well. Similarly, the time-reversed costate dynamics (3.9) have an equilibrium point at the origin when $v(t) = \mathbb{0}_n$, regardless of $x(t)$ and $u(t)$, and (due to Theorem 3.5) they are strongly contracting with rate $c > 0$. Again, we can use Lemma 3.1 to compare $\lambda^{\leftarrow}(t)$ with the trajectory at the origin:

$$\|\lambda^{\leftarrow}(t)\|_\star \leq \|\lambda^{\leftarrow}(0)\|_\star + \frac{1}{c}(1 - e^{-cT}) \sup_{\tau \in [0,T]} \|v(\tau)\|_\star$$

for all $t \in [0, T]$. Since V is bounded, $\lambda^{\leftarrow}(t)$ is confined to a ball Λ about the origin.

3.7.3 Proof of Theorem 3.7

As in Theorem 3.5, let $\lambda^{\leftarrow}(t) = \lambda(T - t)$, so that

$$\frac{d\lambda^{\leftarrow}(t)}{dt} = D_x f(T - t, x(T - t), u(T - t))^\mathsf{T} \lambda^{\leftarrow}(t) - v(t - T)$$

At any fixed t, the $\lambda^{\leftarrow}$ vector field has the Jacobian matrix $D_x f(T - t, x(T - t), u(T - t))^\mathsf{T}$, which is transpose the Jacobian matrix of $f(T - t, \cdot, u(T - t))$. By Assumption 3.1, $\mathrm{osL}(f(T - t, \cdot, u(T - t))) \leq -c$, so Lemma 3.4 implies that the $\lambda^{\leftarrow}$ vector field is also one-sided Lipschitz with constant c, with respect to $\|\cdot\|_\star$. Then we apply Lemma 3.1 to bound $\|\lambda^{\leftarrow}(t) - \bar{\lambda}^{\leftarrow}(t)\|_\star$ with respect to the inputs $u(t)$, $x(t)$, and $v(t)$, resulting in the following bound on $\|\lambda(t)-\bar{\lambda}(t)\|_\star$:

$$\begin{aligned}\|\lambda(t)-\bar{\lambda}(t)\|_\star \leq e^{-c(T-t)}&\|\lambda(T)-\bar{\lambda}(T)\|_\star\\ &+\ell_{f_x,u}\int_t^T e^{-c(\tau-t)}\|u(\tau)-\bar{u}(\tau)\|_U\ d\tau\\ &+\ell_{f_x,x}\int_t^T e^{-c(\tau-t)}\|x(\tau)-\bar{x}(\tau)\|_X\ d\tau\\ &+\int_t^T e^{-c(\tau-t)}\|v(\tau)-\bar{v}(\tau)\|_\star\ d\tau\end{aligned}$$

We apply Lemma 3.1 once more to remove explicit dependence on x, via the bound

$$\int_t^T e^{-c(\tau-t)}\|x(\tau)-\bar{x}(\tau)\|_X\ d\tau \leq \ell_{f,u}\int_t^T\int_0^\tau e^{-c(\tau-t)}e^{-c(\tau-\tau')}\|u(\tau')-\bar{u}(\tau')\|_U\ d\tau d\tau'$$

We then swap the order of integration:

$$\begin{aligned}&\int_t^T\int_0^\tau e^{-c(\tau-t)}e^{-c(\tau-\tau')}\|u(\tau')-\bar{u}(\tau')\|_U d\tau' d\tau\\ &\qquad=\int_0^t\int_t^T e^{-c(\tau-t)}e^{-c(\tau-\tau')}\|u(\tau')-\bar{u}(\tau')\|_U d\tau d\tau'\\ &\qquad\quad+\int_t^T\int_{\tau'}^T e^{-c(\tau-t)}e^{-c(\tau-\tau')}\|u(\tau')-\bar{u}(\tau')\|_U d\tau d\tau'\\ &\qquad=\frac{\sinh(c(T-t))}{c}\int_0^t e^{-c(T-\tau)}\|u(\tau)-\bar{u}\tau)\|_U d\tau\\ &\qquad\quad+\frac{e^{-c(T-t)}}{c}\int_t^T \sinh(c(T-\tau))\|u(\tau)-\bar{u}\tau)\|_U d\tau\end{aligned}$$

3.7.4 Proof of Corollary 3.8

The first three terms are obvious upper bounds on the first three terms in (3.10), and

$$\begin{aligned}&\frac{\sinh(c(T-t))}{c}\int_0^t e^{-c(T-\tau)}\,d\tau + \frac{e^{-c(T-t)}}{c}\int_t^T \sinh(c(T-\tau))\,d\tau\\ &= \int_t^T\int_0^\tau e^{-c(\tau-t)}e^{-c(\tau-\tau')}\,d\tau' d\tau \le \kappa\int_t^T e^{-c(\tau-t)}\,d\tau \le \kappa^2\end{aligned}$$

3.7.5 Proof of Theorem 3.9

Let $u, \bar{u} \in \mathcal{U}$, and let $x, \bar{x}$ and $\lambda, \bar{\lambda}$ be the corresponding state and costate trajectories. Then for all $t \in [0, T]$,

$$\begin{aligned}&\|\mathrm{MSA}(u)(t) - \mathrm{MSA}(\bar{u})(t)\|_U\\ &\quad= \|h(t, x(t), \lambda(t)) - h(t, \bar{x}(t), \bar{\lambda}(t))\|_U\\ &\quad\le \ell_{h,x}\|x(t) - \bar{x}(t)\| + \ell_{h,\lambda}\|\lambda(t) - \bar{\lambda}(t)\|_\star\end{aligned} \tag{3.18}$$

The costate dynamics are (3.9) with $v(t) = -\phi_x(t, x(t), u(t))$, which is bounded in $(\mathbb{R}^n, \|\cdot\|_\star)$ by the boundedness of $x(t)$ and $u(t)$ and the Lipschitz continuity of ϕ_x. By Corollary 3.8,

$$\begin{aligned}\|\lambda(t) - \bar{\lambda}(t)\|_\star &\le \|\lambda(T) - \bar{\lambda}(T)\|_\star\\ &+ \kappa \sup_{t\in[0,T]}\|\phi_x(t, x(t), u(t)) - \phi_x(t, \bar{x}(t), \bar{u}(t))\|_\star\\ &+ \left(\ell_{f_x,u}\kappa + \ell_{f_x,x}\ell_{f,u}\kappa^2\right)\sup_{t\in[0,T]}\|u(t) - \bar{u}(t)\|_U,\end{aligned}$$

where

$$
\begin{aligned}
\|\lambda(T) - \bar{\lambda}(T)\|_\star &= \|\psi_x(x(T)) - \psi_x(\bar{x}(T))\|_\star \\
&\leq \ell_{\psi_x,x}\|x(T) - \bar{x}(T)\| \leq \ell_{\psi_x,x} \sup_{t\in[0,T]} \|x(t) - \bar{x}(t)\|
\end{aligned}
$$

and

$$
\begin{aligned}
&\|\phi_x(t, x(t), u(t)) - \phi_x(t, \bar{x}(t), \bar{u}(t))\|_\star \\
&\qquad \leq \ell_{\phi_x,x}\|x(t) - \bar{x}(t)\| + \ell_{\phi_x,u}\|u(t) - \bar{u}(t)\|_U \\
&\qquad \leq \ell_{\phi_x,x} \sup_{t\in[0,T]} \|x(t) - \bar{x}(t)\| + \ell_{\phi_x,u} \sup_{t\in[0,T]} \|u(t) - \bar{u}(t)\|_U
\end{aligned}
$$

As a consequence of Lemma 3.1, $\sup_{t\in[0,T]}\|x(t) - \bar{x}(t)\| \leq \ell_{f,u}\kappa \sup_{t\in[0,T]}\|u(t) - \bar{u}(t)\|_U$, so we simplify

$$
\begin{aligned}
\|\lambda(t) - \bar{\lambda}(t)\|_\star \leq\ & \ell_{\psi_x,x}\ell_{f,u}\kappa \sup_{t\in[0,T]} \|u(t) - \bar{u}(t)\|_U \\
&+ \kappa^2\ell_{\phi_x,x}\ell_{f,u} \sup_{t\in[0,T]} \|u(t) - \bar{u}(t)\|_U \\
&+ \kappa\ell_{\phi_x,u} \sup_{t\in[0,T]} \|u(t) - \bar{u}(t)\|_U \\
&+ (\ell_{f_x,u}\kappa + \ell_{f_x,x}\ell_{f,u}\kappa^2) \sup_{t\in[0,T]} \|u(t) - \bar{u}(t)\|_U.
\end{aligned}
$$

Substituting the state and costate difference bounds into (3.18) completes the proof of statement (i). Then statement (ii) is a standard consequence of the Banach fixed point theorem.

3.7.6 Proof of Corollary 3.10

We first establish that the fixed points of the MSA operator are precisely the controls that satisfy PMP. One direction is obvious: $u^* = \mathrm{MSA}(u^*)$ implies that u^* satisfies PMP. Now suppose that u^* satisfies PMP, and let x^*, λ^* be the corresponding state and costate trajectories. Then $u^*(t) \in \operatorname{argmin}_{u \in U} H(t, x^*(t), \lambda^*(t), u)$ for all $t \in [0, T]$, so the assumption that the Hamiltonian has a unique minimizer implies that $u^*(t) = h(t, x^*(t), \lambda^*(t))$ for all $t \in [0, T]$, and thus $u^* = \mathrm{MSA}(u^*)$.

We then establish that the MSA iteration converges to a unique fixed point $\hat{u}$. For T sufficiently small or c sufficiently large, κ is sufficiently small that $\mathrm{Lip}(\mathrm{MSA}) \leq b_1\kappa + b_2\kappa^2 < 1$, by Theorem 3.9. Then the Banach fixed point theorem establishes that a unique fixed point $\hat{u}$ exists, and that the iteration from any initial guess converges to $\hat{u}$.

Since an optimal control u^* exists, it is a fixed point of MSA, and the fixed point of MSA is unique. Furthermore, if a control u^* satisfies PMP, then it is a fixed point of MSA, and hence is equal to the optimal control.

Part II

Estimation

Chapter 4

High-Order Network Tomography

This chapter was first published in *IEEE / ACM Transactions on Networking* [120].[1]

Many tasks regarding the monitoring, management, and design of communication networks rely on knowledge of the routing topology. However, the standard approach to topology mapping—namely, active probing with traceroutes—relies on cooperation from increasingly non-cooperative routers, leading to missing information. Network tomography, which uses end-to-end measurements of additive link metrics (like delays or log packet loss rates) across monitor paths, is a possible remedy. Network tomography does not require that routers cooperate with traceroute probes, and it has already been used to infer the structure of multicast trees. This paper goes a step further. We provide a tomographic method to infer the underlying routing topology of an arbitrary set of monitor paths using the joint distribution of end-to-end measurements, without making any assumptions on routing behavior. Our approach, called the *Möbius Inference Algorithm* (MIA), uses cumulants of this distribution to quantify high-order interactions among monitor paths, and it applies Möbius inversion to "disentangle" these interac-

[1] ©2022 IEEE. Reprinted, with permission, from Kevin D. Smith, Saber Jafarpour, Ananthram Swami, and Francesco Bullo, *Topology Inference With Multivariate Cumulants: The Möbius Inference Algorithm*, April 2022.

tions. In addition to MIA, we provide a more practical variant called *Sparse Möbius Inference*, which uses various sparsity heuristics to reduce the number and order of cumulants required to be estimated. We show the viability of our approach using synthetic case studies based on real-world ISP topologies.

4.1 Introduction

Many tasks regarding the monitoring, management, and design of communication networks benefit from the network operator's ability to determine the routing topology, i.e., the incidence between paths and links in the network. During small-scale network failures, for example, routes may automatically switch, and it is important that the network operator has knowledge of the new routing matrix. In the case of large-scale topology failures, inference of the routing topology is a crucial prelude to determining both the surviving network topology and the available services that remain. Peer-to-peer file-sharing networks are another example: nodes may want to know the routing topology so that they can select routes that have minimal overlap with existing routes, so as to avoid congestion and improve performance. Furthermore, the problem of optimal monitor placement relies on some knowledge of the network topology, and inference of the routing matrix provides topological information that could be used to bootstrap new end-to-end measurements.

Literature Review Two main approaches are available for topology inference in communication networks: using *traceroutes*, and using *network tomography* [141]. Traceroutes are the simplest and most direct approach, but they rely on intermediate routers to cooperate by responding to traceroute packets. This cooperation is becoming increasingly uncommon [58], leading to inaccuracies in traceroute-based topology mapping [90].

Some authors have modified traceroute approaches to account for uncooperative routers [137, 72, 63], using partial traceroute results to over-estimate the topology, then applying heuristics and side information to merge nodes. These approaches perform well on test cases, but a rigorous method of selection among viable topologies would still be desirable.

Another approach to topology inference has started to emerge from the literature on network tomography. Network tomography is the problem of inferring additive link metrics (like delays or log packet loss rates) from end-to-end measurements; a nice review is provided in [28]. Unlike traceroute approaches, network tomography does not rely on intermediate routers to cooperate with traces. Instead, it measures some *metric* like delay or log packet loss rate between hosts, and it solves a linear inverse problem to infer the values of these metrics on each link. While most tomography literature assumes that the routing matrix is known, some authors have used tomographic approaches to infer the routing topology in special cases. In general, these approaches are based on a collection of statistics called *path sharing metrics* (PSMs), which are defined for each pair of host-to-host paths. The PSM for a pair of paths is the sum of metrics across all links that are shared by the two paths. A topology is then selected that explains all of the PSMs.

The tomographic approach was first applied to the single-source and multiple-receiver setting to infer multicast trees. One of the first papers to adopt this idea is [104], which uses joint statistics of packet loss between pairs of receivers as a PSM. By repeatedly identifying the pair with greatest path sharing, joining that pair into a "macro-node," and re-computing the statistics, the authors iteratively build the multicast tree from the bottom up. A few years later, [42] generalized this idea from packet losses to other PSMs, including correlations between packet delays between receiver pairs; and [27] accounted for measurement noise by moving the problem to a maximum likelihood framework. Somewhat more recently, [98] re-considered the problem of constructing a multicast tree from PSMs and provided new rigorous and more-efficient algorithms. These papers all reconstruct the tree from PSMs of source-receiver paths.

Later work has extended tomographic topology inference from beyond multicast trees to more general multiple-source, multiple-receiver problems. In [102], the authors merge multicast trees to infer the topology with multiple sources, under some “shortest-path” assumptions on the routing behavior—again using PSMs. [9] provides more general necessary and sufficient conditions for when network inference is possible based on PSMs. Both of these papers essentially assume shortest-path routing, an assumption which is not always valid, for example, due to load balancing in the TCP layer [102]. This assumption also cannot accommodate more complex probing paths, such as the two-way paths that emerge when a monitoring endpoint pings another node.

Recent papers have also applied tomography to problems with uncertain (yet not completely unknown) topologies. In [91], the typical linear inverse problem from tomography is replaced with a Boolean linear inverse problem, allowing the authors to identify failed links from end-to-end data. Similarly, [49] studies the problem of making network tomography robust to dynamics in the network topology. The last two papers also deal with the problem of measurement design, i.e. constructing the routing matrix to ensure identifiability. Neither of these two last papers is concerned with inferring the routing matrix; however, they do represent approaches outside of the PSM paradigm for gleaning topological information from end-to-end data in a tomography setting.

Another recent paper [109] introduced a new method for topology inference, called “OCCAM”. Like most of the other methods we have referenced, OCCAM is based on PSMs; however, instead of algorithmically constructing the unique topology that is consistent with the PSMs and routing assumptions, OCCAM solves an optimization problem with an Occam’s razor heuristic. The heuristic is not guaranteed to find the correct network structure (unless the underlying network is a tree), but the authors demonstrate good empirical performance. To our knowledge, OCCAM is the only approach to truly general topology inference via network tomography, i.e., an approach that does not require any assumptions on routing behavior (beyond the fundamental assumption of stable paths between source-receiver pairs).

Contributions This chapter provides another such approach to topology inference. We extend the use of second-order PSMs into higher-order statistics (i.e., statistics involving more than two paths), allowing us to relax any underlying assumptions about the underlying topology. Our method uses cumulants to quantify high-order interactions between multiple paths, then applies Möbius inversion to "disentangle" these interactions, resulting in an encoding of the routing topology. Our general approach, which we call the *Möbius Inference Algorithm* (MIA), is a non-parametric method of reconstructing the routing matrix from multivariate cumulants of end-to-end measurements, under mild assumptions. It does not require any prior knowledge of the topology or distributions of link metrics, and works under general routing topologies.

The chapter has three main contributions. First, we provide a novel application of statistics and combinatorics to network tomography. We show that multivariate cumulants of end-to-end measurements reveal interactions between the monitor paths (in the form of overlapping links), and we demonstrate how Möbius inversion can be used to infer link-path incidence from these cumulants. Based on these observations, we construct the *Möbius Inference Algorithm* (MIA), which recovers a provably correct routing matrix from these cumulants.

Second, we adapt MIA to the more practical scenario in which a dataset of end-to-end measurements is available, instead of exact cumulants. This "empirical" variant of the routing inference algorithm applies a hypothesis test to every candidate column of the routing matrix, deciding based on the data whether or not the column is present. This hypothesis testing is based on a novel statistic, and it works within any framework for location testing the mean of a distribution.

Third, we create a more practical procedure, called *Sparse Möbius Inference*, which modifies MIA using several sparsity heuristics. This procedure minimizes the number of cumulants that need to be evaluated, restricts cumulant orders to some user-specified limit, and reduces the time complexity of the algorithm. It also makes the inference more robust against measurement noise, by replacing the exact Möbius inversion formula with a lasso regression problem.

Finally, we use many numerical case studies, based on real-world Rocketfuel networks, to evaluate the performance of Sparse Möbius Inference. We study how the performance depends on the underlying network, the number of monitor paths, the sample size, and other parameters.

Organization This chapter takes a didactic approach to introducing MIA and its sparse variant. Section 4.2 formally describes the communication network model and key variables, provides a brief introduction to cumulants and k-statistics, and discusses our three mild assumptions. Section 4.3 considers the easiest setting for topology inference, wherein precise values for all of the necessary cumulants are available without noise, so that we can focus on the core statistical and combinatorial insights behind MIA. Section 4.4 then replaces the precise cumulant values with noisy measurements. Then Section 4.5 replaces MIA altogether with the more practical Sparse Möbius Inference procedure, which allows the user to cap the order of cumulants they are willing to estimate. Finally, Section 4.6 provides an overview of our numerical results and evaluation. Proofs are contained in Section 4.8, and additional numerical results are available in the supplementary material of our publication in *IEEE/ACM Transactions on Networking* [120].

4.2 Modeling and Preliminaries

4.2.1 Model

We consider a network on a (possibly directed) graph G with a set of links $L = \{\ell_1, \ell_2, \ldots, \ell_m\}$. Every link is associated with an additive link metric, like a time delay or log packet loss rate. We will refer to these metrics simply as "delays," although other metrics are possible.

For each link, there is a *link delay variable* U_ℓ, which is a random variable representing the amount of time that a unit of traffic requires to traverse the link. Link delays are not measured directly. Instead, we will infer properties of these variables from cumulative delays across certain simple paths in G, called *monitor paths*. Let P_m be a set of n monitor paths. Each $p \in P_m$ is associated with a *path delay variable*

$$V_p = \sum_{\substack{\ell \in L \text{ s.t.} \\ p \text{ traverses } \ell}} U_\ell, \qquad \forall p \in P_m \tag{4.1}$$

which is the total delay experienced by a unit of traffic along the path p. If we define a random vector of link variables $\mathbf{U} = \begin{pmatrix} U_{\ell_1} & U_{\ell_2} & \cdots & U_{\ell_m} \end{pmatrix}^{\mathsf{T}}$ and a random vector of path variables $\mathbf{V} = \begin{pmatrix} V_{p_1} & V_{p_2} & \cdots & V_{p_n} \end{pmatrix}$, then we can write (4.1) in the form

$$\mathbf{V} = \mathbf{R}\mathbf{U} \tag{4.2}$$

using a *routing matrix* $\mathbf{R} \in \{0, 1\}^{n \times m}$, where $r_{p\ell} = 1$ if and only if p traverses the link ℓ. We stress that we do not make any assumptions about the nature of these monitor paths or the underlying routing behavior. They may be one-way paths between monitoring endpoints, two-way paths from a ping to a node and back, or both. The paths do not have to reflect shortest-path routing.

We suppose that an experimenter is capable of measuring path delays $V_p(t)$ for each monitor path p, at many sample times t. The experimenter has no prior knowledge about the link variables U_ℓ and does not know the routing matrix $\mathbf{R}$. Importantly, we make the simplifying assumption that link delays are spatially and temporally independent, i.e., $U_\ell(t)$ and $U_{\ell'}(t')$ are statistically independent unless $\ell = \ell'$ and $t = t'$. This assumption is fundamental in the network tomography literature [141, 28, 42, 27, 98, 102].

4.2.2 Preliminaries and Notation

General Notation Let $\mathbb{Z}_{\geq 0}$ and $\mathbb{Z}_{>0}$ denote the sets of non-negative and positive integers, respectively. Given a set S and an integer $i \leq |S|$, let the binomial $\binom{S}{i} = \{S' \subseteq S : |S'| = i\}$ denote the collection of all i-element subsets of S. Given $i, n \in \mathbb{Z}_{\geq 0}$, let $\left(\!\binom{n}{i}\!\right)$ denote the number of i-element multisets chosen from n distinct elements. Given two ordered and countable sets $X \subseteq Y$, define the *characteristic vector* $\chi(X, Y) \in \{0, 1\}^{|Y|}$ *of* X *in* Y by $\chi_i(X, Y) = 1$ if and only if $y_i \in X$. Given any function $f : X \to \mathbb{R}$, the *support* of the function $\operatorname{supp}(f)$ is the subset of elements $x \in X$ such that $f(x) \neq 0$.

Multi-Indices A *multiset* is a set that allows for repeated elements. A multiset can be represented by a *multi-index*, which is a function $\alpha : S \to \mathbb{Z}_{\geq 0}$ that maps each element of S to its multiplicity in the multiset. The *support* of a multi-index is the set of elements with positive multiplicity, i.e., $\operatorname{supp}(\alpha) = \{s \in S : \alpha(s) \geq 1\}$. The *size* of a multi-index is its total multiplicity: $|\alpha| = \sum_{s \in S} \alpha(s)$. If S is an ordered set with n elements (e.g., if S consists of elements of a vector), then multi-indices on S are naturally represented as vectors $\alpha \in \mathbb{Z}^n_{\geq 0}$; in this case, we will use multi-indices on S and vectors in $\mathbb{Z}^n_{\geq 0}$ interchangeably. For example, for $S = \{a, b, c, d\}$, the multi-index corresponding to the multiset $\{a, b, b, d, d, d\}$ can be represented by the vector $\begin{pmatrix} 1 & 2 & 0 & 3 \end{pmatrix}^\mathsf{T}$, using an alphabetic ordering of S.

Link Sets Throughout this chapter, we make use of two maps from sets of monitor paths to sets of links. Recall that $\mathbf{R} \in \{0,1\}^{n\times m}$ is the routing matrix. For each $P \subseteq P_m$, we define the *common link set* $C : 2^{P_m} \to 2^L$ by

$$C(P) = \{\ell \in L : r_{p\ell} = 1,\ \forall p \in P\} \tag{4.3}$$

and the *exact link set* $E : 2^{P_m} \to 2^L$ by

$$E(P) = \{\ell \in L : r_{p\ell} = 1,\ \forall p \in P \text{ and } r_{p\ell} = 0,\ \forall p \notin P\} \tag{4.4}$$

The common link set $C(P)$ contains all links that are utilized by every path in P. The exact link set is more strict: $E(P)$ consists of links that are utilized by every path in P *and* that are not utilized by any path outside of P. Neither of these maps are known *a priori*. It is worth noting that the exact link set contains all of the information of the routing matrix, since $E(P)$ is nonempty if and only if the characteristic vector $\chi(P, P_m)$ is a column of $\mathbf{R}$.

As an example, consider the following routing matrix encoding 8 monitor paths that utilize 8 links:

$$\mathbf{R} = \begin{pmatrix} 1&0&0&0&0&0&0&0\\ 0&0&0&0&1&1&0&0\\ 0&0&0&1&0&1&0&1\\ 0&0&1&1&0&1&1&0\\ 0&1&1&1&0&0&0&0\\ 0&1&0&0&1&0&1&0\\ 0&1&1&0&1&0&0&1\\ 0&1&0&0&0&1&1&0 \end{pmatrix}$$

Distribution	Parameters	Cumulants
Normal	$\mu,\ \sigma^2$	$\kappa_1 = \mu,\ \kappa_2 = \sigma^2,\ \kappa_i = 0$ for $i \geq 3$
Exponential	λ	$\kappa_i = \lambda^i(i-1)!$ for $i \geq 1$
Gamma	$\alpha,\ \beta$	$\kappa_i = \alpha\beta^{-i}(i-1)!$ for $i \geq 1$

Table 4.1: Cumulants of common univariate distributions.

In this example, $C(\{p_1\}) = E(\{p_1\}) = \{\ell_1\}$, since column 1 is the only column with a nonzero first entry, and all other entries in the column are zero. Furthermore, $C(\{p_3, p_7\}) = E(\{p_3, p_7\}) = \{\ell_8\}$, since column 8 is the only column with a nonzero third and seventh entry, and all other entries are zero. But C and E are not always equal: $C(\{p_5, p_6\}) = \{p_2\}$, but column 2 contains other nonzero entries as well, so $E(\{p_5, p_6\}) = \emptyset$. Multiple common links are also possible, e.g., $C(\{p_6, p_7\}) = \{\ell_2, \ell_5\}$.

4.2.3 Cumulants and k-Statistics

Cumulants are a class of statistical moments, which extend the familiar notions of mean and covariance to higher orders. A good introduction is provided in [94]; we provide a quick background here. Given a random variable X, define the *cumulant generating function*

$$K(t) = \log \mathrm{E}[e^{tX}] = \kappa_1 t + \frac{\kappa_2}{2!}t^2 + \frac{\kappa_3}{3!}t^3 + \cdots$$

which admits a Taylor expansion for some sequence of coefficients $\kappa_1, \kappa_2, \kappa_3, \ldots$. These coefficients are defined as the *cumulants* of the random variable X. The first three cumulants are identical to central moments: κ_1 is the mean of X, κ_2 is the variance, and $\kappa_3 = \mathrm{E}[(X - \mathrm{E}[X])^3]$. For orders four and higher, the relationship between cumulants and central moments is increasingly complicated. Table 4.1 provides some examples of common distributions whose cumulants have closed-form expressions. Given a random variable X and an integer $i \in \mathbb{Z}_{>0}$, we let $\kappa_i(X)$ denote the ith cumulant of X.

Multivariate cumulants extend cumulants to joint distributions. Given some jointlydistributed random variables $X_1, X_2, \ldots, X_n$, the cumulant generating function is

$$K(\mathbf{t}) = \log \mathrm{E}[e^{t_1 X_1 + \cdots + t_n X_n}] = \sum_{\alpha} \frac{\kappa_\alpha}{|\alpha|!} \mathbf{t}^\alpha$$

where the sum in the Taylor expansion occurs over all multi-indices α on the set of integers $\{1, 2, \ldots, n\}$, and $\mathbf{t}^\alpha$ denotes the product $t_1^{\alpha(1)} t_2^{\alpha(2)} \cdots t_n^{\alpha(n)}$. Collecting $X_1, X_2, \ldots, X_n$ into the random vector $\mathbf{X} = \begin{pmatrix} X_1 & X_2 & \cdots & X_n \end{pmatrix}^\intercal$, we use either the compact notation $\kappa_\alpha(\mathbf{X})$ or expanded notation $\kappa_\alpha(X_1, X_2, \ldots, X_n)$ to represent the multivariate cumulant of the joint distribution that corresponds to the multi-index α. If α is the multi-index of all ones, we drop the subscript and use the shorthand notation $\kappa(X_1, X_2, \ldots, X_n)$. The *order* of a cumulant as the size $|\alpha|$ of its multi-index.

First-order multivariate cumulants are means: if α has all zero multiplicites except $\alpha(i) = 1$, then $\kappa_\alpha(\mathbf{X}) = \mathrm{E}[X_i]$. Second-order multivariate cumulants are covariances: if α has all zero multiplicities except $\alpha(i) = \alpha(j) = 1$, then $\kappa_\alpha(\mathbf{X}) = \mathrm{cov}(X_i, X_j)$. If instead $\alpha(i) = 2$ with all other multiplicities zero, then $\kappa_\alpha(\mathbf{X}) = \mathrm{Var}(X_i)$. We also make use of two general properties of multivariate cumulants:

(i) *Multilinearity.* If Y is a random variable independent from $X_1, X_2, \ldots, X_n$, then

$$\begin{aligned} &\kappa_\alpha(X_1, \ldots, X_i + Y, \ldots, X_n) = \\ &\qquad \kappa_\alpha(X_1, \ldots, X_i, \ldots, X_n) + \kappa_\alpha(X_1, \ldots, Y, \ldots, X_n) \end{aligned}$$

for any index i and multi-index α.

(ii) *Independence.* If any pair X_i, X_j of the random variables $X_1, X_2, \ldots, X_n$ are independent, and $\alpha(i)$ and $\alpha(j)$ are both non-zero, then $\kappa_\alpha(\mathbf{X}) = 0$.

Cumulants can be computed analytically from joint distributions using the generating function, but for unknown distributions, they must be estimated from samples. Given an i.i.d. sample $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N \in \mathbb{R}^n$ from $\mathbf{X}$, the *k-statistic* $k_\alpha(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ is defined as the minimum-variance unbiased estimator of $\kappa_\alpha(\mathbf{X})$. The first and second-order k-statistics are sample means and sample covariances, but higher-order k-statistics quickly become more complex. We refer the reader to [97] and [115] for a discussion of how general k-statistics are derived. For the purpose of this chapter, it suffices to note that software packages are available to compute k-statistics from samples, both in R [96] and our own Python library [114].

4.2.4 Assumptions

At various points throughout the chapter, we will invoke three closely-related assumptions regarding the routing matrix and link delay cumulants. The first assumption requires that $\mathbf{R}$ has no repeated columns:

Assumption 4.1 (Distinct Links)**.** *No two links are traversed by precisely the same set of paths in P_m; i.e., no two columns of $\mathbf{R}$ are identical; i.e., $|E(P)| \in \{0,1\}$ for all $P \subseteq P_m$.*

This assumption is common in the network tomography literature. If $\ell, \ell' \in L$ are used by precisely the same set of monitor paths, then the link delays $U_\ell, U_{\ell'}$ will only show up in path delays through their sum $U_\ell + U_{\ell'}$. Due to this linear dependence, complete network tomography is impossible when Assumption 4.1 is violated, since $\mathbf{R}$ will be rank deficient.

The second assumption requires that link delays have nonzero cumulants:

Assumption 4.2 (Nonzero Cumulants)**.** *For all $\ell \in L$, and for all $i = 2, \ldots, n$, the delay cumulant is nonzero: $\kappa_i(U_\ell) \neq 0$.*

For most practical purposes, one can think of Assumption 4.2 as meaning that no link delay distribution is normally distributed. Non-normality is a necessary condition for the assumption to hold, since the normal distribution has zero-valued cumulants for orders 3 and higher. Non-normality is not technically a sufficient condition, since it is theoretically possible for a distribution to have zero cumulants at some orders, but these cases are rare. In fact, the normal distribution is the only distribution with a finite number of nonzero cumulants [94]. If link delays are known to be non-normally distributed, we consider this to be a weak assumption.

Finally, the third assumption requires that certain *sums* of link delays have nonzero cumulants:

Assumption 4.3 (Nonzero Common Cumulants). *For all* $P \subseteq P_m$*, and for all* $i = 2, 3, \ldots, n$*, if* $C(P)$ *is nonempty, then* $\sum_{\ell \in C(P)} \kappa_i(U_\ell) \neq 0$.

In other words, if all paths in $P \subseteq P_m$ share a collection of common links $C(P)$, the delay cumulants on these common links should not cancel out by summing to zero. This is also a weak assumption, since such a cancellation is very unlikely. In fact, many families of distributions supported on $\mathbb{R}_{>0}$ (including exponential and gamma distributions) have strictly positive cumulants at all orders, in which case Assumption 4.3 is satisfied automatically.

4.3 Theoretical Foundations

We now proceed with our main theoretical contribution: a simple algorithm to infer the routing matrix from multivariate cumulants of path latencies. The purpose of this section is to state the underlying theoretical principles of MIA, so we will temporarily assume that exact values for multivariate cumulants of the path delay vector $\mathbf{V}$ are available. In reality, the experimenter seldom knows these exact values and must estimate them via k-statistics instead, but this requires some extra statistical treatment that we defer to Sections 4.4 and 4.5. For now, we assume exact cumulant values to focus on the discrete mathematics that underpin MIA.

MIA works by identifying which exact link sets $E(P)$ are nonempty, since these correspond precisely to columns of $\mathbf{R}$ (via the characteristic vector of P). The sizes of the exact link sets are not directly observable, but they can be inferred from the sizes of the common link sets. From (4.3) and (4.4), we can see that exact and common link sets are related by

$$E(P) = C(P) \setminus \bigcup_{p' \notin P} C(P \cup \{p'\}).$$

We can count the size of the union using the inclusion-exclusion principle:

$$\left| \bigcup_{p' \notin P} C(P \cup \{p'\}) \right| = \sum_{Q \supset P} (-1)^{|Q|-|P|+1} |C(Q)|. \tag{4.5}$$

Since $C(P \cup \{p'\}) \subseteq C(P)$ for all p', we can use the inclusion-exclusion formula (4.5) to find the size of the exact link set as a function of the sizes of the common link sets:

$$|E(P)| = \sum_{Q \supseteq P} (-1)^{|Q|-|P|} |C(Q)| \tag{4.6}$$

If we could somehow evaluate the number of common links shared by any set of monitor paths, we could use the inclusion-exclusion principle to compute any $|E(P)|$, from which we could reconstruct the routing matrix.

Unfortunately, counting the number of common links is typically infeasible in a tomography setting. But the relationship in (4.6) actually holds for *any* additive measure of link sets, not just cardinality, and some additive measures can be inferred directly from end-to-end path data. For example, if "$|C(Q)|$" represents the sum of delay variances $\mathrm{Var}(U_\ell)$ for each link in $C(Q)$, then (4.6) yields the sum of delay variances across links in $E(P)$, which is nonzero if and only if $E(P)$ is nonempty. This sum of delay variances across common links can be inferred from path delay data—at least for pairs of monitor paths p, p', the covariance $\mathrm{cov}(V_p, V_{p'})$ is equal to the sum of delay variances for each shared link in $C(\{p, p'\})$. For larger path sets, we require higher-order statistics—like multivariate cumulants—to measure "$|C(P)|$".

Having conveyed some of the core ideas behind MIA, we are ready to present the algorithm itself and examine it with more theoretical rigor. The algorithm occurs in three stages:

(i) *Estimation.* Estimate a vector of multivariate cumulants of path latencies. This vector contains information about the links that are common to any given collection of paths. (The label "estimation" is a misnomer in the context of this section, wherein cumulants are known precisely, but it will make more sense when we consider the "data-driven" version of the algorithm.)

(ii) *Inversion.* Apply a Möbius inversion transformation to this vector of estimates. The vector resulting from this transformation contains the routing matrix, under a simple encoding. The transformation is linear, so this step can be viewed as a matrix-vector multiplication.

(iii) *Reconstruction.* Decode the transformed vector, thereby reconstructing the routing matrix.

Theorem 4.1 (Analysis of MIA)**.** *Consider the application of Algorithm 4.1 to a joint distribution of path delays* $\mathbf{V} = \begin{pmatrix} V_{p_1} & V_{p_2} & \cdots & V_{p_n} \end{pmatrix}^\intercal$*. Let* $\mathbf{R} \in \{0,1\}^{n \times m}$ *be the true*

Algorithm 4.1 Möbius Inference Algorithm (MIA)

```
Require: Joint distribution of path delays $\mathbf{V}$
Ensure: Routing matrix $\hat{\mathbf{R}}$
{Estimation stage:}
Initialize undefined function $f_n : 2^{P_m} \to \mathbb{R}$
for $P \subseteq P_m$ do
    Define $\alpha$ as any multi-index on $P_m$ such that $\mathrm{supp}(\alpha) = P$ and $|\alpha| = n$
    $f_n(P) \leftarrow \kappa_\alpha(\mathbf{V})$
end for
{Inversion stage:}
Initialize undefined function $g_n : 2^{P_m} \to \mathbb{R}$
for $P \subseteq P_m$ do
    $g_n(P) \leftarrow \sum_{Q \supseteq P} (-1)^{|Q|-|P|} f_n(Q)$
end for
{Reconstruction stage:}
Initialize empty matrix $\hat{\mathbf{R}} \in \mathbb{R}^{n \times 0}$
for $P \subseteq P_m$ do
    if $g_n(P) \neq 0$ then
        $\hat{\mathbf{R}} \leftarrow \left(\hat{R} \quad \chi(P, P_m)\right)$
    end if
end for
return $\hat{\mathbf{R}}$
```

underlying routing matrix, and let $\mathbf{U} = \begin{pmatrix} U_{\ell_1} & U_{\ell_2} & \cdots & U_{\ell_m} \end{pmatrix}^\mathsf{T}$ *be the underlying link delays, so that* $\mathbf{V} = \mathbf{R}\mathbf{U}$*. The following are true:*

(i) The algorithm terminates and returns a matrix $\hat{\mathbf{R}} \in \{0,1\}^{n \times \hat{m}}$ *for some* $\hat{m} \in \mathbb{Z}_{\geq 0}$*, in* $O(2^n)$ *time.*

(ii) By line 7, the map $f_n : 2^{P_m} \to \mathbb{R}$ *satisfies the following property:*

$$f_n(P) = \sum_{\ell \in C(P)} \kappa_n(U_\ell), \qquad \forall P \subseteq P_m \tag{4.7}$$

(iii) By line 12, the map $g_n : 2^{P_m} \to \mathbb{R}$ *satisfies the following property:*

$$g_n(P) = \sum_{\ell \in E(P)} \kappa_n(U_\ell), \qquad \forall P \subseteq P_m \tag{4.8}$$

(iv) Every column of $\hat{\mathbf{R}}$ *is also a column of* $\mathbf{R}$*. Furthermore, under Assumptions 4.1 and 4.2,* $\mathbf{R}$ *and* $\hat{\mathbf{R}}$ *are equivalent (up to a permutation of columns).*

Statement (i) is obvious from inspection of the algorithm, so we will focus on proving the remaining three statements, which fall neatly into the three stages (estimation, inversion, and reconstruction) of the algorithm. In the following subsections, we will analyze each of these three stages.

4.3.1 Estimation Stage

The purpose of the estimation stage is to collect a vector of high-order statistics of path delays. These statistics are carefully chosen so that they contain information about the routing topology. The title of "estimation" for this stage will be more appropriate in the next subsection, when we must estimate these statistics from data (rather than compute them analytically from a known distribution).

In the estimation stage, we gather a vector of multivariate path delay cumulants for every path set $P \subseteq P_m$. The multivariate cumulants that we select for each path set are based on representative multi-indices:

Definition 4.2 (Representative Multi-Indices)**.** *Let $P \subseteq P_m$, and let $i \geq |P|$ be an integer. An ith-order representative multi-index of P is any multi-index α on P_m such that $\operatorname{supp}(\alpha) = P$ and $|\alpha| = i$. We use $A_{i,P}$ to denote the set of all ith-order representative multi-indices of P.*

We will now collect a vector of path delay cumulants, with one entry corresponding to each set of monitor paths in 2^{P_m}:

Definition 4.3 (Common Cumulant)**.** *Let i be a positive integer. For each $P \subseteq P_m$, let α be any ith-order representative multi-index of P. The ith-order common cumulant is the map $f_i : 2^{P_m} \to \mathbb{R}$ with entries*

$$f_i(P) = \kappa_\alpha(\mathbf{V}), \qquad \forall P \subseteq P_m \tag{4.9}$$

Careful readers will also note that we refer to “the” common cumulant, rather than “a” common cumulant, which would seem more appropriate, given the many choices of representative multi-indices. But the value of the common cumulant is independent of the particular choice of representative multi-index—regardless of which representative multi-index we choose, it is always the sum of univariate cumulants across links that are traversed by every path in P. Broadly speaking, the value of $f_i(P)$ contains information about which links are common to every path in P.

Lemma 4.4 (Properties of the Estimation Stage)**.** *The following are true:*

(i) *Let $P \subseteq P_m$. If $i \geq |P|$, there are $\binom{i-1}{|P|-1}$ ith-order representative multi-indices of P.*

(ii) *For all $i \in \mathbb{Z}_{>0}$, the common cumulant $f_i : 2^{P_m} \to \mathbb{R}$ satisfies (4.7).*

(iii) *Statement (ii) of Theorem 4.1 is true, i.e., Algorithm 4.1 correctly computes the common cumulant vector for order $i = n$.*

4.3.2 Inversion Stage

In the inversion stage, we extract topological information from the vector of common cumulants by applying an invertible linear transformation. Lemma 4.4 (ii) shows that common cumulants are sums over common link sets. But it is clear from (4.3) and (4.4) that common link sets can be written as unions of exact link sets, which more directly provide information about the routing matrix. Accordingly, common cumulants can be written as sums over exact link sets, using *exact cumulants*:

Definition 4.5 (Exact Cumulant)**.** *For each positive integer i, we define the ith-order exact cumulant $g_i : 2^{P_m} \to \mathbb{R}$ by (5.10), replacing n with i.*

In the following lemma, we formalize the relationship of common cumulants as sums of exact cumulants. We then apply Möbius inversion to this sum:

Lemma 4.6 (Properties of the Inversion Stage)**.** *Let f_i be the common cumulant vector, and let $g_i : 2^{P_m} \to \mathbb{R}$. The following three statements are equivalent:*

(i) *g_i is the exact cumulant vector.*

(ii) *f_i and g_i satisfy*

$$f_i(P) = \sum_{Q \supseteq P} g_i(Q), \qquad \forall P \subseteq P_m \tag{4.10}$$

(iii) *f_i and g_i satisfy*

$$g_i(P) = \sum_{Q \supseteq P} (-1)^{|Q|-|P|} f_i(Q), \qquad \forall P \subseteq P_m \tag{4.11}$$

Furthermore, statement (iii) of Theorem 4.1 is true, i.e., the Algorithm 4.1 correctly computes the exact cumulant vector.

Lemma 4.6 is the heart of MIA. By applying the inversion (4.11) to the vector of common cumulants, we calculate the vector of *exact* cumulants. Whereas common cumulants contain information about which links are traversed by every path in a set, exact cumulants contain information about which links are traversed *precisely* by the paths in a set, i.e., they contain information about columns of the routing matrix.

4.3.3 Reconstruction Stage

The final stage of the algorithm is to reconstruct the routing matrix from the exact cumulant vector. This reconstruction is straightforward, using only the zero-nonzero pattern of g_i:

Lemma 4.7 (Properties of the Reconstruction Stage)**.** *Let $g_n : 2^{P_m} \to \mathbb{R}$ be the exact cumulant vector. For each $P \subseteq P_m$, let $\chi(P, P_m) \in \{0,1\}^n$ be the characteristic vector of P in P_m. The following are true:*

(i) *If $P \in \operatorname{supp}(g_n)$, then $\chi(P, P_m)$ must be a column of the routing matrix. Under Assumptions 4.1 and 4.2, the converse is also true.*

(ii) *Statement (iv) of Theorem 4.1 is true.*

4.3.4 Detailed Example

In order to illustrate MIA, we will apply the algorithm to a small example, consisting of 3 monitor paths that utilize three links. We will walk through each of the three stages of the algorithm in detail.

Setup Consider a network with three monitor paths $P_m = \{p_1, p_2, p_3\}$ and three links $L = \{\ell_1, \ell_2, \ell_3\}$, with a routing matrix

$$\mathbf{R} = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \end{array} \begin{array}{c} \begin{array}{ccc} \ell_1 & \ell_2 & \ell_3 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \end{array} \tag{4.12}$$

Clearly this routing matrix satisfies Assumption 4.1. Each of the three link delay distributions is exponential, with probability density functions $f_{u_\ell}(x) = \lambda_\ell e^{-\lambda_\ell x}$ for each $\ell \in L$, and intensities $\lambda_{\ell_1} = 1$, $\lambda_{\ell_2} = 1.5$, and $\lambda_{\ell_3} = 2$ (in units of per millisecond). All cumulants of exponential distributions are positive, so the latency variables satisfy Assumption 4.2. We then invoke (4.2) to obtain the joint distribution of path delays. We assume that the theoretical distribution of path delays is known—in particular, the cumulants $\kappa_\alpha(\mathbf{V})$ are known exactly—and our objective is to use these cumulants to infer the routing matrix, via Algorithm 4.1.

Estimation Stage

There are seven non-empty subsets of P_m. Sets with one path only have one 3rd-order representative multi-index; for example, the path set $P = \{p_1\}$ has a unique representative multi-index $\alpha = (3, 0, 0)$. Sets with two paths have 2 representative multi-indices;

for example, $P = \{p_1, p_2\}$ has $\alpha = (2,1,0)$ and $\alpha' = (1,2,0)$. The three-element path set $P = P_m$ has only the one representative multi-index $\alpha = (1,1,1)$. For each of these seven path sets, we will select one of the representative multi-indices arbitrarily and collect them into the common cumulant vector. For example:

$$\mathbf{f}_3 = \begin{pmatrix} f_3(\{p_1\}) \\ f_3(\{p_2\}) \\ f_3(\{p_3\}) \\ f_3(\{p_1, p_2\}) \\ f_3(\{p_1, p_3\}) \\ f_3(\{p_2, p_3\}) \\ f_3(P_m) \end{pmatrix} = \begin{pmatrix} \kappa_{(3,0,0)}(\mathbf{V}) \\ \kappa_{(0,3,0)}(\mathbf{V}) \\ \kappa_{(0,0,3)}(\mathbf{V}) \\ \kappa_{(1,2,0)}(\mathbf{V}) \\ \kappa_{(1,0,2)}(\mathbf{V}) \\ \kappa_{(0,1,2)}(\mathbf{V}) \\ \kappa_{(1,1,1)}(\mathbf{V}) \end{pmatrix} = \begin{pmatrix} 70/27 \\ 9/4 \\ 1/4 \\ 2 \\ 0 \\ 1/4 \\ 0 \end{pmatrix}$$

It is worth noting that $\mathbf{f}_3$ agrees with (4.7), i.e., we can decompose the vector into univariate cumulants of link delays:

$$\mathbf{f}_3 = \begin{pmatrix} \kappa_{(3,0,0)}(\mathbf{V}) \\ \kappa_{(0,3,0)}(\mathbf{V}) \\ \kappa_{(0,0,3)}(\mathbf{V}) \\ \kappa_{(1,2,0)}(\mathbf{V}) \\ \kappa_{(1,0,2)}(\mathbf{V}) \\ \kappa_{(0,1,2)}(\mathbf{V}) \\ \kappa_{(1,1,1)}(\mathbf{V}) \end{pmatrix} = \begin{pmatrix} \kappa_3(U_1) + \kappa_3(U_2) \\ \kappa_3(U_1) + \kappa_3(U_3) \\ \kappa_3(U_3) \\ \kappa_3(U_1) \\ 0 \\ \kappa_3(U_3) \\ 0 \end{pmatrix} = \begin{pmatrix} 70/27 \\ 9/4 \\ 1/4 \\ 2 \\ 0 \\ 1/4 \\ 0 \end{pmatrix}$$

Of course, performing this decomposition relies on our prior knowledge of $\mathbf{R}$ and the link delay distributions, which are unavailable to the experimenter.

Inversion Stage

In order to obtain the exact cumulant vector $\mathbf{g}_3$ from the common cumulant vector $\mathbf{f}_3$, we apply the Möbius inversion transformation (4.11). Note that this transformation is linear, and it can be represented in the matrix form $\mathbf{g}_3 = \mathbf{X}\mathbf{f}_3$, where the matrix $\mathbf{X}$ contains the coefficients $(-1)^{|Q|-|P|}$:

$$\begin{pmatrix} g_3(\{p_1\}) \\ g_3(\{p_2\}) \\ g_3(\{p_3\}) \\ g_3(\{p_1,p_2\}) \\ g_3(\{p_1,p_3\}) \\ g_3(\{p_2,p_3\}) \\ g_3(P_m) \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & -1 & -1 & 0 & 1 \\ 0 & 1 & 0 & -1 & 0 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{X}} \begin{pmatrix} f_3(\{p_1\}) \\ f_3(\{p_2\}) \\ f_3(\{p_3\}) \\ f_3(\{p_1,p_2\}) \\ f_3(\{p_1,p_3\}) \\ f_3(\{p_2,p_3\}) \\ f_3(P_m) \end{pmatrix}$$

Evaluating this transformation, we obtain the following expression for the exact cumulant vector:

$$\mathbf{g}_3 = \begin{pmatrix} g_3(\{p_1\}) \\ g_3(\{p_2\}) \\ g_3(\{p_3\}) \\ g_3(\{p_1,p_2\}) \\ g_3(\{p_1,p_3\}) \\ g_3(\{p_2,p_3\}) \\ g_3(P_m) \end{pmatrix} = \begin{pmatrix} 16/27 \\ 0 \\ 0 \\ 2 \\ 0 \\ 1/4 \\ 0 \end{pmatrix}$$

We can verify that these values for $\mathbf{g}_3$ agree with both (5.10) and (4.10). For example, the routing matrix (4.12) implies that $E(\{p_1\}) = \{\ell_2\}$, so (5.10) gives

$$g_3(\{p_1\}) = \frac{2}{\lambda_{\ell_2}^3} = \frac{16}{27}$$

in agreement with our computed result for $\mathbf{g}_3$. Furthermore, (4.10) claims that we can decompose $f_3(\{p_1\})$ according to

$$\begin{aligned} f_3(\{p_1\}) &= g_3(\{p_1\}) + g_3(\{p_1, p_2\}) + g_3(\{p_1, p_3\}) + g_3(P_m) \\ &= \frac{70}{27} \end{aligned}$$

in agreement with $f_3(\{p_1\})$ from the previous stage.

Reconstruction Stage

All that remains is to examine the zero-nonzero pattern of $\mathbf{g}_3$. Note that $\mathbf{g}_3$ has three non-zero entries: $P_1 = \{p_1\}$, $P_2 = \{p_1, p_2\}$, and $P_3 = \{p_2, p_3\}$. We can then reconstruct the routing matrix from the characteristic vectors of these three path sets:

$$\hat{\mathbf{R}} = \begin{pmatrix} \chi(P_1, P_m) & \chi(P_2, P_m) & \chi(P_3, P_m) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

Observe that $\hat{\mathbf{R}}$ is equivalent to the "ground truth" routing matrix in (4.12), modulo an irrelevant permutation of columns, as guaranteed by Theorem 4.1 (iv).

4.4 From Distributions to Data

Having presented the core theory underlying MIA, we now turn to a more practical problem: routing matrix inference from data, rather than from a theoretical distribution. Instead of knowing the joint distribution of the path delay vector $\mathbf{V}$, in this section, we only assume that an i.i.d. sample $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N \in \mathbb{R}^n$ of is available. Thus, instead of using ground-truth cumulant values $\kappa_\alpha(\mathbf{V})$ in the estimation stage of the algorithm, we have to use estimates of these cumulants via the k-statistics $k_\alpha(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N)$. Moreover, because k-statistics introduce noise into the inference procedure, we will also need to modify the reconstruction stage to be robust against this noise.

Estimation Stage In lines 4 and 5 of Algorithm 4.1, MIA selects an arbitrary representative multi-index $\alpha \in A_{n,P}$ and records the common cumulant value $f_n(P) \leftarrow \kappa_\alpha(\mathbf{V})$. The choice of representative multi-index here is truly arbitrary, since all yield an identical value for $\kappa_\alpha(\mathbf{V})$. This is not true for k-statistics. While the expected values of $k_\alpha(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N)$ are identical for all $\alpha \in A_{n,P}$, the actual values of these statistics will generally be different. It is not clear that any of these values is a better estimate than the others, so we propose replacing $\kappa_\alpha(\mathbf{V})$ with the average

$$\hat{f}_n(P) = \binom{n-1}{|P|-1}^{-1} \sum_{\alpha \in A_{n,P}} k_\alpha(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N) \tag{4.13}$$

of all k-statistics for the representative multi-indices of P. Thus, we replace both lines 4 and 5 in Algorithm 4.1 with (4.13), as well as using the notation $\hat{f}_n(P)$ instead of $f_n(P)$ (to highlight that the algorithm is now using an estimate of the common cumulant instead of its true value).

Inversion Stage There is no need to modify the inversion stage of the algorithm in the data-driven setting. The inversion stage simply applies the linear transformation $\mathbf{g}_n = \mathbf{X}\mathbf{f}_n$, where $\mathbf{X}$ encodes the Möbius inversion. When we switch from $\mathbf{g}_n$ and $\mathbf{f}_n$ to vectors of estimates $\hat{\mathbf{g}}_n$ and $\hat{\mathbf{f}}_n$, this transformation is still valid in expectation:

$$\mathrm{E}[\hat{\mathbf{g}}_n] = \mathbf{X}\,\mathrm{E}[\hat{\mathbf{f}}_n] = \mathbf{X}\mathbf{f}_n = \mathbf{g}_n$$

Reconstruction Stage In line 15 of Algorithm 4.1, MIA checks if an entry of the exact cumulant vector is nonzero. But in the data-driven scenario, we switch from exact cumulants to estimates $\hat{\mathbf{g}}_n$, which only match the zero-nonzero pattern of $\mathbf{g}_n$ in expectation. To account for inevitable noise in these estimates, instead of checking if $\hat{g}_n(P) = 0$, we must adopt some kind of hypothesis test $\texttt{Nonzero}(g_n(P) \mid \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N)$, i.e., some decision rule to guess whether $g_n(P) \neq 0$ based on the data. We will examine the construction of such a test in the next subsection.

The performance of MIA in the data-driven setting depends entirely on the accuracy of the hypothesis test. This accuracy depends on the test itself, the choice of test parameters (like significance levels), and the size of the sample size N, so it is difficult to state general theoretical guarantees regarding the algorithm. Nonetheless, some guarantees are evident in extreme cases, if Assumptions 4.1 and 4.2 are satisfied:

(i) If the test has no Type I error, i.e., if $g_n(P) = 0$ always leads to a decision that $\texttt{Nonzero}(g_n(P) \mid \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N)$ is false, then every column of $\hat{\mathbf{R}}$ will be a true column of $\mathbf{R}$.

(ii) If the test has no Type II error, then $\hat{\mathbf{R}}$ will contain every column of $\mathbf{R}$.

(iii) If the test is *consistent*, in the sense that the test is free of both Type I and Type II error in $N \to \infty$ limit, then similarly $\hat{\mathbf{R}} = \mathbf{R}$ in the $N \to \infty$ limit.

For all practical purposes, none of these extreme cases will apply, and we will have to rely on the algorithm's performance in test scenarios to assess its usefulness.

4.4.1 Hypothesis Tests

We now examine the hypothesis test $\texttt{Nonzero}(g_n(P) \mid \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N)$, which we will subsequently abbreviate as $\texttt{Nonzero}(g_n(P))$. Because $\mathrm{E}[\hat{g}_n(P)] = g_n(P)$, we can assess the null hypothesis $g_n(P) = 0$ via an equivalent null hypothesis, that $\mathrm{E}[\hat{g}_n(P)] = 0$. There is no single correct way to perform this mean location test—many approaches exist, with advantages and disadvantages.

Normal Approximation

Because the statistics $\hat{g}_n(P)$ are asymptotically normally distributed, we could simply estimate the mean and variance of the distribution and apply a standard z-test. This approach is used in [125], for example, to perform hypothesis testing on univariate cumulants, using univariate k-statistics. Unfortunately, while the mean of the distribution is easily estimated by $\hat{g}_n(P)$, the variance relies on computing variances of multivariate k-statistics, which are both mathematically and computationally complex.

Sample Splitting

Another simple approach is to partition the original N-length sample into M subsamples of size N/M, compute $\hat{g}_n(P)$ for each subsample, and use standard hypothesis testing to assess whether the statistics have zero mean. Since the subsamples are non-overlapping, each of the M values of $\hat{g}_n(P)$ will be iid, so standard approaches (like the 1-sample Student's t-test [36, §9.5]) can be used to test the null hypothesis that $\mathrm{E}[\hat{g}_n(P)] = 0$.

Bootstrapping

Bootstrapping (see, e.g., [33, Chapter 2]) is a resampling technique that uses the empirical distribution (i.e., the discrete distribution with uniform weight on each sample value) to approximate the original distribution. For $b = 1, 2, \ldots, M$ (where typically $M \approx 50$), we define a *resample* $\tilde{\mathbf{v}}_{b1}, \tilde{\mathbf{v}}_{b2}, \ldots, \tilde{\mathbf{v}}_{bN}$ that is chosen randomly with replacement from the original sample $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N$. We then compute $\hat{g}_n(P)$ for each resample, resulting in a sample of size M for $\hat{g}_n(P)$, which we can use to perform a mean hypothesis test. This approach has been applied to estimating confidence intervals for cumulants [142].

4.4.2 Detailed Example

In order to illustrate the empirical version of MIA, we will continue to use the low-dimensional example from Section 4.3.4, with the same routing matrix (4.12) and the same exponentially-distributed link delays. We created a synthetic dataset with 900 independent samples from each link distribution, which we transformed into 900 samples of V_{p_1}, V_{p_2} , and V_{p_3} based on the sums encoded in $\mathbf{R}$.

We use the sample splitting approach to the $\texttt{Nonzero}(g(P))$ hypothesis test in this example. The 900 original sample points are split into 30 samples of size 30. To carry out the estimation stage, we estimate the common cumulant vector for each of these 30

P	$f_3(P)$	$\hat{f}_{3,P}$	$g_3(P)$	$\hat{g}_{3,P}$
$\{p_1\}$	2.59	2.67 ± 0.5	0.593	0.66 ± 0.2
$\{p_2\}$	2.25	2.31 ± 0.7	0	0.06 ± 0.2
$\{p_3\}$	0.25	0.24 ± 0.05	0	0.02 ± 0.02
$\{p_1, p_2\}$	2	2.01 ± 0.6	2	2.01 ± 0.5
$\{p_1, p_3\}$	0	-0.01 ± 0.05	0	-0.01 ± 0.04
$\{p_2, p_3\}$	0.25	0.23 ± 0.07	0.25	0.23 ± 0.06
$\{p_1, p_2, p_3\}$	0	0.00 ± 0.09	0	0.00 ± 0.09

Table 4.2: Common and exact cumulants in the low-dimensional example. Columns $f_3(P)$ and $g_3(P)$ report the true underlying values, while $\hat{f}_3(P)$ and $\hat{g}_3(P)$ show the mean and standard error of the respective estimates.

samples with the simple average of k-statistics in (4.13):

$$\hat{\mathbf{f}}_3 = \begin{pmatrix} \hat{f}_3(\{p_1\}) \\ \hat{f}_3(\{p_2\}) \\ \hat{f}_3(\{p_3\}) \\ \hat{f}_3(\{p_1, p_2\}) \\ \hat{f}_3(\{p_1, p_3\}) \\ \hat{f}_3(\{p_2, p_3\}) \\ \hat{f}_3(P_m) \end{pmatrix} = \begin{pmatrix} k_{(3,0,0)}(\cdot) \\ k_{(0,3,0)}(\cdot) \\ k_{(0,0,3)}(\cdot) \\ \frac{1}{2}k_{(1,2,0)}(\cdot) + \frac{1}{2}k_{(2,1,0)}(\cdot) \\ \frac{1}{2}k_{(1,0,2)}(\cdot) + \frac{1}{2}k_{(2,0,1)}(\cdot) \\ \frac{1}{2}k_{(0,1,2)}(\cdot) + \frac{1}{2}k_{(0,2,1)}(\cdot) \\ k_{(1,1,1)}(\cdot) \end{pmatrix}$$

Here $k_\alpha(\cdot)$ is shorthand for $k_\alpha(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N)$. Columns 2 and 3 of Table 4.2 report the means and standard errors for these 30 estimates of $\hat{\mathbf{f}}_3$. To perform the inversion stage, the the vector $\hat{\mathbf{g}}_3$ is then computed by $\hat{\mathbf{g}}_3 = \mathbf{X}\hat{\mathbf{f}}_3$, where $\mathbf{X}$ is the matrix defined in Section 4.3.4. Columns 4 and 5 of Table 4.2 similarly summarize the distribution of these 30 estimates for $\hat{\mathbf{g}_3}$. Indeed, all of the $\hat{f}_3(P)$ and $\hat{g}_3(P)$ averages are within one standard error of $f_3(P)$ and $g_3(P)$, respectively.

Based on these 30 estimates of $\hat{\mathbf{g}}_3$, we perform the reconstruction stage using a 1-sample Student's t-test to assess the null hypothesis that $\mathrm{E}[\hat{g}_3(P)] = 0$ for each path set. The p-value for each null hypothesis is reported in Table 4.3, as well as the result of the

P	p-value for $g_3(P) = 0$	$\chi(P)$ is in R?
$\{p_1\}$	0.001	Yes
$\{p_2\}$	0.8	No
$\{p_3\}$	0.5	No
$\{p_1, p_2\}$	0.0005	Yes
$\{p_1, p_3\}$	0.9	No
$\{p_2, p_3\}$	0.0008	Yes
$\{p_1, p_2, p_3\}$	1	No

Table 4.3: Hypothesis testing for whether or not $\chi(P, P_m)$ is a column of the routing matrix, at 0.01 significance.

test at a significance of 0.01.

For precisely three of the path sets, we reject the null hypothesis that $g_3(P) = 0$: $P_1 = \{p_1\}$, $P_2 = \{p_1, p_2\}$, and $P_3 = \{p_2, p_3\}$. Assembling the characteristic vectors of these path sets into $\hat{\mathbf{R}}$, we obtain an identical estimate to our result from Section 4.3.4, which is identical to the ground truth routing matrix (up to a permutation of columns).

4.5 Sparse Möbius Inference

The key step in the Möbius Inference Algorithm is the linear transformation $\mathbf{g}_i = \mathbf{X}\mathbf{f}_i$, where $\mathbf{g}_i$ is a vector of $2^n - 1$ exact cumulants, $\mathbf{f}_i$ is a vector of $2^n - 1$ common cumulants, n is the number of monitor paths, and $\mathbf{X}$ is the matrix encoding Möbius inversion. Three problems arise naturally: the computational expense of the transformation $\mathbf{X}$, the impracticality of populating every entry of $\mathbf{f}_i$ with empirical measurements, and the noise present in $\mathbf{f}_i$ (and $\mathbf{g}_i$) due to the use of cumulants with excessively high order. In this section, we simultaneously tackle these three problems using several different sparsity heuristics.

Our proposed "Sparse Möbius Inference" procedure proceeds in three stages. In the first stage, we use measurements of low-order common cumulants to identify which entries of the $\mathbf{f}_i$ and $\mathbf{g}_i$ vectors can contain nonzero entries. We can then ignore all other entries of these vectors and drop their corresponding columns and rows from $\mathbf{X}$, reducing the Möbius inversion down to a (typically much) smaller set of equations. In the second stage, we impose the following sparsity heuristic on $\mathbf{g}_i$: if P is a sufficiently large path set that is strictly contained within some other path set in $\operatorname{supp}(f_i)$, then $g_i(P) = 0$. This heuristic allows us to remove further entries from both $\mathbf{g}_i$ and $\mathbf{f}_i$, provided we make a suitable modification to $\mathbf{X}$. Finally, in the third stage, we apply a sparsity-promoting lasso optimization problem to filter noisy estimates of common cumulants and impute the values of common cumulants that are impractical to measure. The end result is a sparse estimate for $\mathbf{g}_i$, which only relies on estimates of common cumulants up to a small, user-specified order.

4.5.1 Stage 1: Bound the Support of f_i

In the first stage, we estimate the collection of path sets $P \subseteq P_m$ for which $f_i(P) \neq 0$. The key to this process is the observation that $f_i(Q) \neq 0$ only if $f_i(P) \neq 0$ for all subsets $P \subseteq Q$: if just a single subset P has a zero-valued common cumulant, then $C(P) = \emptyset$, which implies that $C(Q) = \emptyset$. If we focus on small path sets, then we can use low-order cumulants to identify which of these path sets have no common links, and remove all of their supersets from the support of $\mathbf{f}_i$.

We can maintain a compact representation of our estimate of $\operatorname{supp}(\mathbf{f}_i)$ using a *bounding topology*. A bounding topology is any collection of path sets $\mathcal{B} \subseteq 2^{P_m}$ with the following property: if $f_i(P) \neq 0$, then $\mathcal{B}$ contains some path set $B \in \mathcal{B}$ such that $P \subseteq B$. We will refer to the collection of all sets contained by some $B \in \mathcal{B}$ (i.e., the union $\bigcup_{B\in\mathcal{B}} 2^B$) as the "support estimate" of $\mathcal{B}$. Below are two extreme examples:

- $\mathcal{B} = \{P_m\}$ is trivially a bounding topology, albeit not a very informative one, since the support estimate is 2^{P_m}.

- $\mathcal{B} = \mathrm{supp}(\mathbf{g}_i)$ is a bounding topology: if $f_i(P) \neq 0$, then some superset $B \supseteq P$ satisfies $g_i(B) \neq 0$, and thus $B \in \mathcal{B}$. This is a "tight" bounding topology, in the sense that every set in its support estimate is indeed in the support of $\mathbf{f}_i$.

Stage 1 begins with an uninformative bounding topology (like $\mathcal{B} = \{P_m\}$), and it iteratively "tightens" $\mathcal{B}$ using successive orders of common cumulant estimates. The fundamental idea is that if we determine $\texttt{Nonzero}(f_i(P))$ is false for some small path set P, then we ought to split up all $B \in \mathcal{B}$ containing P into smaller sets that do not contain P, thereby eliminating all supersets of P from the support estimate. This iterative tightening procedure then terminates at a (typically small) user-specified cumulant order.

Unfortunately, $\texttt{Nonzero}(f_i(P))$ is usually a hypothesis test with limited statistical power—there is a chance that our data would incorrectly indicate that $f_i(P) = 0$, leading us to remove any superset of P from the support estimate and thus ignore nonzero values of the common cumulant in future calculations. Such an error could greatly harm the accuracy of later stages of the topology inference. In order to hedge against this possibility, we propose a robust procedure that splits a set $B \in \mathcal{B}$ only if a sufficient number of subsets of B are found to have zero common cumulant. The user provides a *threshold function* $t : \mathbb{Z}_{>0} \times \mathbb{Z}_{>0} \to \mathbb{Z}_{>0}$, where $B \in \mathcal{B}$ is never split so long as $t(|B|, i)$ size-i subsets of $|B|$ are found to have a nonzero common cumulant.

The core of the procedure is Algorithm 4.2, which tightens an estimate of the bounding topology using common cumulants of some fixed order i. The algorithm initially computes the collection of all size-i sets P in the support estimate of $\mathcal{B}$ for which $\texttt{Nonzero}(f_i(P))$ is true. What follows is effectively a voting procedure: each of these sets P counts as a "vote" in favor of keeping each superset $Q \supseteq P$ in the support estimate. If one of the sets $B \in \mathcal{B}$ fails to reach its threshold of $t(|B|, i)$ votes, then B is split up into the $|B|$ subsets obtained by removing one element from B, and the votes for these subsets are tallied as well. This process repeats until all the sets in $\mathcal{B}$ with size at least i reach their respective thresholds. Theorem 4.8 formally states the guarantees of this algorithm:

Algorithm 4.2 `Tighten`$(\mathcal{B}, i, t)$

```
Require: Bounding topology $\mathcal{B} \subseteq 2^{P_m}$, cumulant order $i \in \mathbb{Z}_{>0}$, and threshold function
    $t : \mathbb{Z}_{>0} \times \mathbb{Z}_{>0} \to \mathbb{Z}_{>0}$
Ensure: Tightened bounding topology $\mathcal{B}' \subseteq 2^{P_m}$
Initialize $\mathcal{B}' = \emptyset$, $\mathcal{X} = \emptyset$, and
    $$\mathcal{P} = \left\{ P \in \bigcup_{B \in \mathcal{B}} \binom{B}{i} : \texttt{Nonzero}(f_i(P)) \right\}$$
while $|\mathcal{B}| > 0$ do
    Remove an arbitrary set $B$ from $\mathcal{B}$ and add it to $\mathcal{X}$
    if $|B| < i$ or $|\{P \in \mathcal{P} : P \subseteq B\}| \geq t(|B|, i)$ then
        $\mathcal{B}' \leftarrow \mathcal{B}' \cup \{B\}$
    else
        for $p \in B$ do
            $B_{\text{sub}} \leftarrow B \setminus \{p\}$
            if $B_{\text{sub}} \notin \mathcal{X}$ and no set in $\mathcal{B} \cup \mathcal{B}'$ contains $B_{\text{sub}}$ then
                $\mathcal{B} \leftarrow \mathcal{B} \cup \{B_{\text{sub}}\}$
            end if
        end for
    end if
end while
return $\mathcal{B}'$
```

Theorem 4.8 (Properties of Algorithm 4.2)**.** *Let $\mathcal{B} \subseteq 2^{P_m}$ be a collection of path sets, let $i \in \mathbb{Z}_{>0}$ be a cumulant order, and let $t : \mathbb{Z}_{>0} \times \mathbb{Z}_{>0} \to \mathbb{Z}_{>0}$ be a threshold function. The following are true:*

(i) Algorithm 4.2 evaluates `IsNonzero`$(f_i(P))$ $O(n^i)$ times and terminates after $O(2^q)$ iterations of the while loop, where q is the size of the largest set in $\mathcal{B}$. The algorithm returns a collection of path sets $\mathcal{B}' \subseteq 2^{P_m}$.

(ii) The support estimate of $\mathcal{B}'$ is a subset of the support estimate of $\mathcal{B}$.

(iii) For any set P in the support estimate of $\mathcal{B}$, P is also in the support estimate of $\mathcal{B}'$ only if either $|P| < i$, or if there is a superset $Q \supseteq P$ in the support estimate of $\mathcal{B}$ for which at least $t(|Q|, i)$ size-i subsets $R \subseteq Q$ satisfy `Nonzero`$(f_i(R))$.

Through the repeated application of Algorithm 4.2 to a collection $\mathcal{B}$ and successively larger orders i, as detailed in Algorithm 4.3, we obtain tighter support estimates. Every path set in $\mathrm{supp}(\mathbf{f}_i)$ should remain in the support estimate of $\mathcal{B}$ after each iteration, so long as the values of the threshold function t are sufficiently small (and the test `Nonzero`$(f_i(P))$ is sufficiently accurate). Furthermore, as we incorporate information from higher-order cumulants, we remove path sets for which $f_i(P) = 0$ from the support estimate. In summary, the support estimate of $\mathcal{B}$ becomes a more and more accurate approximation of $\mathrm{supp}(\mathbf{f}_i)$.

Algorithm 4.3 `BoundingTopology`$(\mathcal{B}, i_0, i_f, t)$

```
Require: Initial guess $\mathcal{B} \subseteq 2^{P_m}$, initial cumulant order $i_0$, final cumulant order $i_f$, and threshold function $t : \mathbb{Z}_{>0} \times \mathbb{Z}_{>0} \to \mathbb{Z}_{>0}$
Ensure: Tightened bounding topology $\mathcal{B} \subseteq 2^{P_m}$
for $i = i_0, i_0 + 1, \ldots, i_f$ do
    $\mathcal{B} \leftarrow$ Tighten$(\mathcal{B}, i, t)$
end for
return $\mathcal{B}$
```

We will conclude the discussion of Stage 1 by addressing two questions—how should we select the initial guess for $\mathcal{B}$ that is supplied to Algorithm 4.3, and how should we design the threshold function t?

Choosing an Initial Bounding Topology A safe (albeit inefficient) choice for the initial guess of bounding topology is $\mathcal{B} = \{2^{P_m}\}$. Clearly the support estimate of $\mathcal{B}$ will contain every path set in $\mathrm{supp}(\mathbf{f}_i)$. Unfortunately, this choice also maximizes the runtime of Algorithm 4.3, since the sub-routine Algorithm 4.2 is exponential in the size of the largest set in $\mathcal{B}$.

A more practical approach is to use second-order cumulants (i.e., covariances) to construct an initial guess for $\mathcal{B}$. Second-order k-statistics tend to have a small variance (compared to the higher-order k-statistics), leading to only a small probability that $\texttt{Nonzero}(f_2(P))$ yields a false negative, which makes the thresholding in Algorithm 4.2 unnecessary. If we require that $\texttt{Nonzero}(f_2(P))$ is true for *all* two-element subsets of each set in $\mathcal{B}$, then we can use second-order cumulants to construct a more efficient initial guess for $\mathcal{B}$, and then we can run Algorithm 4.3 on this initial guess starting at order $i_0 = 3$.

One way to efficiently construct this covariance-based initial guess is to use standard algorithms for maximal clique enumeration. Recall from graph theory that a *clique* is any set of nodes for which all nodes in the set are adjacent, and a *maximal clique* is a clique that is not contained within a larger clique. Construct a graph $G_b = (P_m, E_b)$ where each monitor path is a node, and an edge $\{p_i, p_j\}$ is included in E_b if and only if $\texttt{Nonzero}(f_2(\{p_i, p_j\}))$ is true. Cliques in G_b are precisely the path sets for which $\texttt{Nonzero}(f_2(P))$ is true of every two-element subset. Therefore, we take as our initial guess for $\mathcal{B}$ the set of maximal cliques in G_b. The size of the largest clique is typically significantly smaller than n, leading to a faster runtime for Algorithm 4.3.

Constructing the Threshold Function Algorithm 4.3 requires the user to specify a threshold function $t(|P|, i)$, indicating the minimum number of size-i subsets of P that must pass the nonzero common cumulant test for P to remain in the support estimate. Choosing the threshold value is a balance—large values may lead to sets in $\text{supp}(f_i)$ being rejected from the support estimate, but small values will cause information from many zero-valued cumulants to be ignored. We will try to devise an intuitive and tunable form for $t(|P|, i)$ to strike this balance.

Recall that the *statistical power* of a hypothesis test is the probability of rejecting the null hypothesis given that the alternative hypothesis is true—in our case, the probability that $\texttt{Nonzero}(f_i(P))$ is true if indeed $P \in \operatorname{supp}(\mathbf{f}_i)$. Suppose that, for each $P \in \operatorname{supp}(\mathbf{f}_i)$, the corresponding test $\texttt{Nonzero}(f_i(P))$ is true independently and with uniform probability $1 - \beta$. Under these (inaccurate but nonetheless useful) assumptions, the number of size-i subsets of any $Q \in \operatorname{supp}(\mathbf{f}_i)$ for which $\texttt{Nonzero}(f_i(P))$ is true follows a binomial distribution, with $\binom{|Q|}{i}$ trials and a success probability of $1 - \beta$. Hence, the probability that at least $t(|Q|, i)$ size-i subsets of Q pass the nonzero test is $1 - F_{|Q|,i}(t(|Q|, i))$, where $F_{|Q|,i}$ is the cdf of the binomial distribution.

Because Q truly belongs to the support of $\mathbf{f}_i$, it is highly undesirable that we erroneously remove Q from the support estimate by setting the threshold $t(|Q|, i)$ inappropriately high. To render such an error unlikely, we must ensure that $1 - F_{|Q|,i}(t(|Q|, i))$ exceeds some high probability $1 - \gamma \in (0, 1)$, e.g., $1 - \gamma = 0.1$. Once we specify γ, we can solve for the appropriate threshold as the quantity

$$\begin{aligned} t(|Q|, i) &= \max\{t \in \mathbb{Z}_{>0} : F_{|Q|,i}(t) < \gamma\} \\ &= \min\{t \in \mathbb{Z}_{>0} : F_{|Q|,i}(t) \geq \gamma\} - 1 \end{aligned}$$

In other words, we set $t(|Q|, i)$ as one less the γ quantile of the binomial distribution with $\binom{|Q|}{i}$ trials and success probability $1 - \beta$. There is no good closed-form expression for the value of this quantile; however, it is readily computable in many statistics packages.

This binomial quantile specification for $t(|Q|, i)$ is somewhat informal, since the outcomes of $\texttt{Nonzero}(f_i(P))$ are neither independently nor identically distributed, as the derivation assumed. However, the method does at least provide an intuitive way to reduce the specification of t down to two tunable parameters, $\gamma \in (0, 1)$ (the highest tolerable probability that $Q \in \operatorname{supp}(\mathbf{f}_i)$ is accidentally rejected) and $\beta \in (0, 1)$ (an estimate for the probability that $\texttt{Nonzero}(f_i(P))$ yields a false negative). We could also specify different values of these parameters for different k-statistic orders i, to account for the fact that k-statistics tend to become less accurate with higher orders.

4.5.2 Stage 2: Bound the Support of g_i

In the previous stage, we used information from low-order cumulants to narrow the entries of $\mathbf{f}_i$ containing nonzero entries down to the support estimate of $\mathcal{B}$. Because $f_i(P) = 0$ implies that $g_i(P) = 0$ as well, this stage also simultaneously restricts the nonzero entries of $\mathbf{g}$ to the support estimate of $\mathcal{B}$. The second stage drops even more zero-valued entries from these two vectors. Instead of using empirical information from low-order cumulants, this stage enforces a “hard” sparsity heuristic: that $g_i(P) = 0$ for all path sets P larger than some threshold size s, unless that path set is an element of $\mathcal{B}$. In other words, we assume that the only “large” path sets are those contained directly in the bounding topology inferred from low-order cumulants.

This heuristic immediately zeros out large swaths of the $\mathbf{g}_i$ vector, allowing us to ignore them during the final stage. But the heuristic also allows us to drop even more entries from the $\mathbf{f}_i$ vector, as stated in the following lemma:

Lemma 4.9 (Elimination of Large, Non-Maximal Path Sets)**.** *Let $\mathcal{B} \subseteq 2^{P_m}$ be a collection of path sets, and let $s \in \mathbb{Z}_{>0}$. Assume that the following are true:*

(i) Every set in $\mathcal{B}$ is maximal (i.e., no $B, B' \in \mathcal{B}$ exist such that $B \subset B'$),

(ii) $f_i(P) \neq 0$ and $g_i(P) \neq 0$ only if P is in the support estimate of $\mathcal{B}$, and

(iii) $g_i(P) = 0$ for all $P \subseteq P_m$ with $|P| > s$ and $P \notin \mathcal{B}$.

Then for every P in the support estimate of $\mathcal{B}$ such that $|P| \leq s$,

$$\begin{aligned} g_i(P) = &\sum_{Q \supseteq P : |Q| \leq s} (-1)^{|Q|-|P|} f_i(Q) \\ &- \sum_{B \in \mathcal{B} : B \supseteq P} (-1)^{s-|P|} \binom{|B| - |P| - 1}{s - |P|} f_i(B) \end{aligned} \tag{4.14}$$

Due to (4.14), there is no need to measure or keep track of $f_i(P)$ for sufficiently large P, unless P is a set in $\mathcal{B}$. Note that these common cumulants are not just zeroed out—they take on a nonzero value; however, this value is constrained to a linear combination of the common cumulants for $B \in \mathcal{B}$, which are already elements of the common cumulant vector.

4.5.3 Stage 3: Lasso Optimization

The previous two stages eliminated large parts of the $\mathbf{f}_i$ and $\mathbf{g}_i$ vectors, using a combination of information from low-order cumulants, *a priori* assumptions, and suitable modifications of the Möbius transformation matrix $\mathbf{X}$. These two stages significantly reduce the computational expense of performing Möbius inversion and populating $\mathbf{f}_i$ with empirical estimates of common cumulants. Furthermore, because the first stage tends to eliminate the largest subsets of P_m from the support for $\mathbf{f}_i$, we can populate $\mathbf{f}_i$ with cumulants of order lower than n. But this cumulant order (which must be at least the size of the largest path set with a nonzero common cumulant) can still be unrealistically large, and the resulting common cumulant estimates can be quite noisy. In the final stage of Sparse Möbius Inference, we address these two problems by filtering $\mathbf{f}_i$ using lasso optimization.

To set up the problem, the user first supplies a maximum cumulant order $i_{\max} \in \mathbb{Z}_{>0}$, indicating the largest order of cumulant they are willing to estimate. Based on $i_{\max}$, we partition the common cumulant vector by $\mathbf{f}_{i_{\max}} = \begin{pmatrix} \mathbf{f}_o & \mathbf{f}_u \end{pmatrix}^{\mathsf{T}}$, and we make the corresponding partition to the inversion matrix $\mathbf{X} = \begin{pmatrix} \mathbf{X}_o & \mathbf{X}_u \end{pmatrix}$. $\mathbf{f}_o$ corresponds to the common cumulants $f_{i_{\max}}(P)$ of path sets with size at most $i_{\max}$, i.e., the common cumulants that we can "observe" using empirical estimates. All other "unobserved" common cumulants are consigned to the $\mathbf{f}_u$ vector. Note that $\mathbf{f}_o$ is not directly populated with common cumulant estimates: in fact, both $\mathbf{f}_o, \mathbf{f}_u$ are left as decision variables in the lasso optimization problem, and the value of $\mathbf{f}_o$ is allowed to deviate from the empirical estimate if it promotes a sparser solution $\mathbf{g}$. Instead, all of the empirical common cumulant estimates are collected into a vector $\hat{\mathbf{f}}_o$, and the corresponding standard deviations of each estimate are collected into the vector σ. We then solve for the optimal common cumulant vector $\mathbf{f}^* = \begin{pmatrix} \mathbf{f}_o^* & \mathbf{f}_u^* \end{pmatrix}^{\mathsf{T}}$ using the convex, unconstrained optimization problem:

$$\begin{aligned} \mathbf{f}_o^*, \mathbf{f}_u^* &= \underset{\mathbf{f}_o, \mathbf{f}_u}{\operatorname{argmin}}\, J(\mathbf{f}_o, \mathbf{f}_u) \\ J(\mathbf{f}_o, \mathbf{f}_u) &= ||\mathbf{\Sigma}^{-1}(\mathbf{f}_o - \hat{\mathbf{f}}_o)||_2^2 + ||\mathbf{D}(\mathbf{X}_o\mathbf{f}_o + \mathbf{X}_u\mathbf{f}_u)||_1 \end{aligned} \tag{4.15}$$

Here $\mathbf{\Sigma} = \operatorname{diag}\{\sigma\}$, and $\mathbf{D}$ is some tunable diagonal matrix of positive weights (which we will soon discuss in more detail). Having computed the solution, we then evaluate $\mathbf{g}^* = \mathbf{X}_o\mathbf{f}_o^* + \mathbf{X}_u\mathbf{f}_u^*$.

Eqn. (5.13) simultaneously de-noises measurements of the observed common cumulant values and imputes the unobserved common cumulants. The quadratic term is proportional to the log likelihood of the data $\hat{\mathbf{f}}_o$ (under the assumption of independent and normally-distributed common cumulant estimates with variances σ^2), and the regularizer $||\mathbf{X}_o\mathbf{f}_o + \mathbf{X}_u\mathbf{f}_u||_1$ encourages sparsity in the vector $\mathbf{g}^*$. The end result is an estimate of $\mathbf{g}_{i_{\max}}$ that only measures common cumulants up to a user-specified order and is more robust to noise in these measurements.

As with the full Möbius Inference Algorithm, the columns of the routing matrix correspond to the nonzero entries of $\mathbf{g}_{i_{\max}}$. Thus, once we obtain an optimal (and sparse) exact cumulant vector $\mathbf{g}^*$, we add the characteristic vector of each $P \in \operatorname{supp}(\mathbf{g}^*)$ to our estimate of $\mathbf{R}$.

Weighting the 1-Norm A straightforward choice for weighting the 1-norm of $\mathbf{g}^*$ is to choose a uniform weighting strategy, in which case $\mathbf{D} = \lambda \mathbf{I}$ for some parameter $\lambda > 0$ that weights the 1-norm relative to the log likelihood of the data. But uniform weighting tends to suppress entries of $\mathbf{g}^*$ corresponding to singleton path sets. If $P = \{p\}$ for some $p \in P_m$, then (4.14) shows that $g_i(P)$ is the only entry of $\mathbf{g}_i$ that depends on $f_i(P)$. Thus, if the uncertainty σ in the measurement of $\hat{f}_o(P)$ is sufficiently large, the optimizer is free to zero out $g^*(P)$ by tuning the decision variable corresponding to $f_i(P)$. Indeed, we have observed numerically that uniform weighting leads to routing matrix estimates missing many columns with single nonzero entries.

To counteract this problem, we suggest applying less weight to "under-determined" entries of $\mathbf{g}^*$. Formally, for each P in the support estimate, let

$$a(P) = \begin{cases} \left|\{Q \text{ in supp. est.} : \mathbf{X}_o(Q,P) > 0\}\right|, & |P| \leq i_{\max} \\ \left|\{Q \text{ in supp. est.} : \mathbf{X}_u(Q,P) > 0\}\right|, & |P| > i_{\max} \end{cases}$$

be the number of entries of $\mathbf{g}^*$ that depend on the decision variable corresponding to $f_{i_{\max}}(P)$. We then choose the weight corresponding to $g^*(P)$ according to $d(P) = \lambda a(P)^b$, where $\lambda > 0$ is a uniform overall weight for the 1-norm term, and $b \in [0,1)$ is some exponent. The exponent should be non-negative to ensure that the weight is increasing in $a(P)$, but it should also be fairly small, so that the weight's rate of change rapidly tapers off for positive $a(P)$. We have found empirically that setting b between 0.2 and

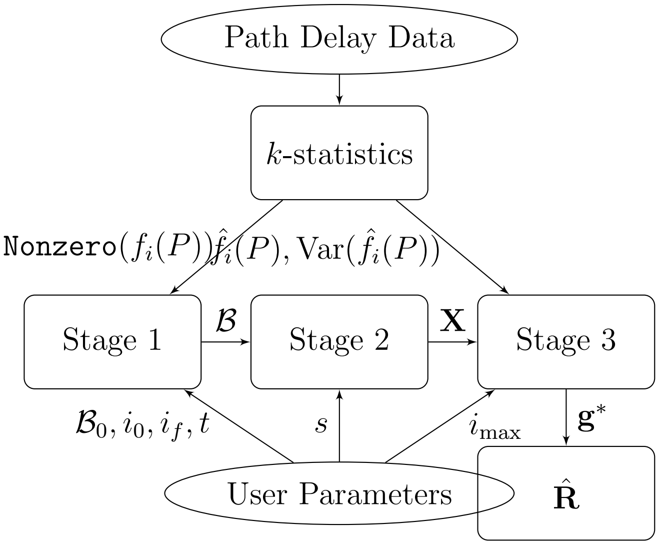

Figure 4.1: Diagram of the Sparse Möbius Inference procedure.

0.4 is generally a good choice.

4.5.4 Putting Everything Together

For completeness, we now show how the three stages of the Sparse Möbius Inference procedure come together to form a data-to-routing-matrix pipeline. Figure 5.1 depicts a diagram of this process.

The user begins Stage 1 with an initial guess of the bounding topology $\mathcal{B}_0 \subseteq 2^{P_m}$ (either $\{P_m\}$ or maximal cliques of the graph formed by nonzero covariances), an initial cumulant order i_0 (usually 2 or 3), a final cumulant order i_f (e.g., 4 or 5), and a threshold function t (perhaps using quantiles of the binomial distribution). Algorithm 4.3 then tightens the support estimate by setting $\mathcal{B} = \texttt{BoundingTopology}(\mathcal{B}_0, i_0, i_f, t)$, using the path delay dataset to evaluate $\texttt{Nonzero}(f_i(P))$ for orders $i = i_0, i_0 + 1, \ldots, i_f$. Then $\mathcal{B}$ is passed on to Stage 2.

In the second stage, the user provides a size threshold s for the "hard" sparsity heuristic. In accordance with (4.14), the modified Möbius inversion matrix $\mathbf{X}$ is constructed,

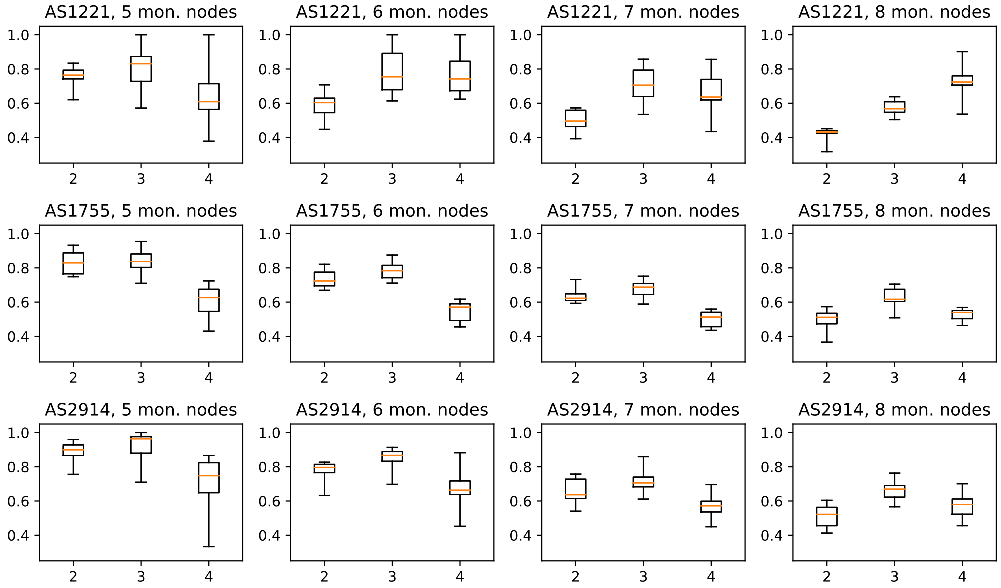

Figure 4.2: Distributions of F1 scores of the routing matrix estimate for the 120 case studies, based on a sample of size 100,000. Plots in each row are based on the same underlying network, and plots in the same column have the same number of monitor nodes. The three boxes in each plot correspond to values $i_{\max} = 2, 3, 4$ used for inference.

considering only rows and columns of the matrix corresponding to path sets in the support estimate of $\mathcal{B}$ that are either directly in $\mathcal{B}$ or at most of size s. This matrix $\mathbf{X}$ is passed to Stage 3.

To begin the final stage, the user specifies a cumulant order $i_{\max}$ (e.g., 3, 4, or 5) and partitions the common cumulant vector and the matrix $\mathbf{X}$ accordingly. For path sets of size at most $i_{\max}$, the path delay data is once again used to estimate the common cumulants $\hat{\mathbf{f}}_o$ and the variances σ^2 of these estimates. Solving (5.13) yields a filtered common cumulant vector $\mathbf{f}^*$, leading to a sparse estimate $\mathbf{g}^* = \mathbf{X}\mathbf{f}^*$ of the exact cumulant vector. Finally, the estimate $\hat{\mathbf{R}}$ is constructed from the zero-nonzero pattern of $\mathbf{g}^*$.

4.6 Results and Evaluation

What follows is an abbreviated set of experimental results applying Sparse Möbius Inference to many synthetic datasets. The full description of our methodology and results are contained in the supplementary material of our publication in *IEEE/ACM Transactions on Networking* [120]. We use the F1 score as an accuracy metric, which is the harmonic mean of the precision (positive predictive value) and recall (detection probability) of a binary classifier. Our implementation of the Sparse Möbius Inference procedure is available at https://github.com/KevinDalySmith/high-order-tomography.

Synthetic Datasets We created 120 synthetic datasets based on real ISP network topologies, provided by Rocketfuel [124]. We selected three networks within the Rocketfuel database with different sizes and densities (AS1221, AS1755, and AS2914). For each topology, we generated 40 synthetic datasets of path delays: 10 each for experiments with 5, 6, 7, and 8 monitor nodes. For each of these 40 case studies, the network links are assigned different gamma delay distributions, the n_{node} monitor nodes are selected at random, and the $n = \binom{n_{\text{node}}}{2}$ monitor paths are chosen by computing the shortest path between each pair of monitor nodes. Then a large sample of the joint path delay distribution is recorded.

Sparsity of the Common and Exact Cumulants The Sparse Mobius Inference procedure is based on the postulate that the vectors of common and exact cumulants are both sparse. This assumption holds up extremely well in our case studies; with $n = 28$ paths, for example, 99.99% to 99.999% of the entries of the common cumulant vector are zero.

Evaluating the Bounding Topology The first stage of Sparse Möbius Inference uses low-order cumulants to estimate $\mathrm{supp}(\mathbf{f}_i)$. Our results indicate that Algorithm 4.3 is very effective at finding a bounding topology with a tight support estimate. For almost all of the 120 case studies, third-order cumulants ($i_f = 3$) with a sample size $N = 50,000$ or larger are sufficient to construct a bounding topology that predicts $\mathrm{supp}(\mathbf{f}_i)$ with an F1 score of 1.0 (or extremely close to 1.0).

Evaluating the Estimated Routing Matrix Next, we evaluate the performance of Sparse Möbius Inference end-to-end. We ran stages 2 and 3 to get an estimate of $\hat{\mathbf{R}}$ for each case study and various sample sizes, using as input to Stage 2 the bounding topologies computed with $i_f = 4$ from the same sample. The hyperparameters of the lasso heuristic (λ and the exponent b) are tuned separately for each underlying network and number of monitor paths. Figure 4.2 shows the F1 scores that we obtained for each of the 120 case studies. For all underlying networks, the performance tends to degrade with the number of monitor paths, and the best estimate is usually obtained using third-order k-statistics ($i_{\max} = 3$).

Evaluating the Lasso Heuristic We also evaluated the lasso heuristic in Stage 3 using ground-truth cumulants. For these experiments, we borrowed the bounding topologies computed from the $N = 100,000$ sample with $i_f = 4$, but instead of populating the $\hat{\mathbf{f}}_o$ vector in (5.13) with k-statistics computed from this sample, we used the true common cumulants. These values have no uncertainty, so we removed the quadratic penalty from $J(\mathbf{f}_o, \mathbf{f}_u)$, instead constraining $\mathbf{f}_o = \hat{\mathbf{f}}_o$. Again, the hyperparameters λ and b are tuned separately for each network and number of monitor paths. Figure 4.3 plots the distribution of the resulting F1 scores. For smaller (5 or 6 monitor) scenarios, the lasso heuristic typically achieves 100% accurate routing matrix reconstruction. In larger scenarios, the

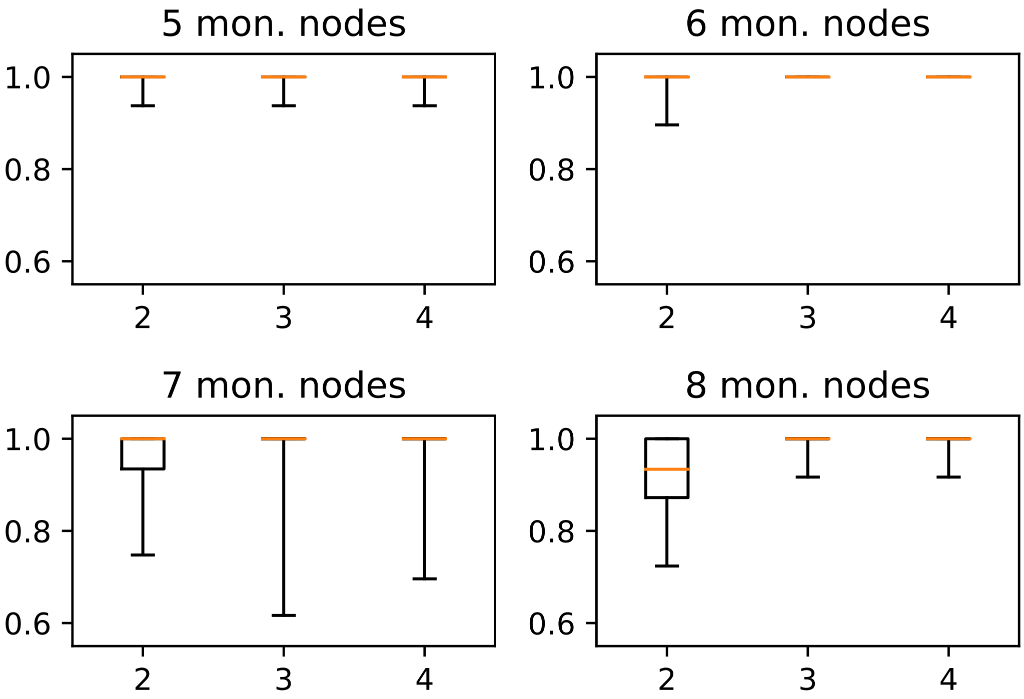

Figure 4.3: Distributions of F1 scores of the routing matrix estimate based on ground-truth cumulants (instead of k-statistics). Each plot corresponds to a particular number of monitor paths, and the results are aggregated across case studies from the 3 underlying networks. The three boxes in each plot correspond to values $i_{\max} = 2, 3, 4$.

heuristic requires up to third-order cumulants for completely accurate inference.

Discussion Our results paint a mixed but optimistic picture for the Sparse Möbius Inference procedure. Admittedly, higher F1 scores from the $N = 100,000$ sample would be desirable before the method is deployed in real-world applications. But the two key components of the procedure—estimating $\text{supp}(\mathbf{f}_i)$ from low-order k-statistics, and using the lasso sparsity heuristic to infer $\mathbf{R}$ without using the high-order cumulants required by MIA—worked well in isolation, achieving 100% accuracy in most scenarios.

4.7 Conclusion

We have provided a novel tomographic approach to routing topology inference from path delay data, without making any assumptions on routing behavior. Through MIA, we have provided a theoretical framework for extending the use of second-order statistics in network tomography toward higher-order statistics. Furthermore, we have introduced the Sparse Möbius Inference procedure, which implements a heuristic and more practical variant of MIA. We have examined the performance of Sparse Möbius Inference using many synthetic case studies. While more work is needed to improve the filtering of noisy k-statistics, our results indicate that the Sparse Möbius Inference can serve as a solid foundation for future improvements.

4.8 Proofs

4.8.1 Proof of Lemma 4.4

To prove (i), we will count the number of ways that i "counts" of multiplicity can be assigned to the support of a representative multi-index. Each element of P contains at least one count, and we are free to distribute the remaining $i - |P|$ counts arbitrarily across the elements of P. Thus, there are $\left(\!\binom{|P|}{i-|P|}\!\right)$ ways to distribute the remaining counts, which is equivalent to $\binom{i-1}{|P|-1}$.

To prove (ii), let α be some ith-order representative multi-index of P. Using the independence of U_ℓ and the multilinearity of multivariate cumulants, we have

$$\begin{aligned} f_i(P) &= \kappa\left(\underbrace{\mathbf{R}^{(1)}\mathbf{U}, \dots, \mathbf{R}^{(1)}\mathbf{U}}_{\alpha(1)\ \text{times}}, \dots, \underbrace{\mathbf{R}^{(n)}\mathbf{U}, \dots, \mathbf{R}^{(n)}\mathbf{U}}_{\alpha(n)\ \text{times}}\right) \\ &= \sum_{\ell=1}^{m} \left(r_{1\ell}^{\alpha(1)} \cdots r_{n\ell}^{\alpha(n)}\right) \kappa\left(\underbrace{U_\ell, \dots, U_\ell}_{\alpha(1)+\cdots+\alpha(n)\ \text{times}}\right) \\ &= \sum_{\ell=1}^{m} \left(\prod_{j\in\operatorname{supp}(\alpha)} r_{j\ell}\right) \kappa_i(U_\ell) \end{aligned}$$

where $\mathbf{R}^{(j)}$ denotes the jth row of $\mathbf{R}$. Since $\prod_{j\in\operatorname{supp}(\alpha)} r_{j\ell} = 1$ if $\ell \in C(P)$ and is zero otherwise, we obtain

$$f_i(P) = \sum_{\ell\in C(P)} \kappa_n(U_\ell), \qquad \forall P \subseteq P_m$$

To prove (iii), observe that the estimation stage of Algorithm 1 defines the map f_n precisely according to Definition 3, so that f_n is the common cumulant vector by line 6 of the algorithm. Then statement (iii) follows by statement (ii) of this lemma.

4.8.2 Proof of Lemma 4.6

We begin with the equivalence (ii) $\iff$ (iii). This equivalence holds for *any* functions $f_i, g_i : 2^{P_m} \to \mathbb{R}$, and it follows from the Möbius inversion formula applied over 2^{P_m}. See, for example, [2, Theorem 5.1].

To prove that (i) $\implies$ (ii), we will first show that

$$C(P) = \bigcup_{Q\supseteq P} E(Q) \tag{4.16}$$

Let $\ell \in C(P)$, and examine the column of the routing matrix $\mathbf{R}_\ell \in \{0,1\}^n$. There is some $Q \subseteq P_m$ for which the characteristic vector satisfies $\chi(Q, P_m) = \mathbf{R}_\ell$. It follows that

$\ell \in E(Q)$. Now, because $\ell \in C(P)$, it follows that $r_{p\ell} = 1$ for all $p \in P$, so that $Q \supseteq P$. Therefore $\ell \in \bigcup_{Q \supseteq P} E(Q)$. Next, let $\ell \in \bigcup_{Q \supseteq P} E(Q)$, so that $\ell \in E(Q)$ for some $Q \supseteq P$. It is clear that $r_{p\ell} = 1$ for all $p \in Q$, so the inclusion $Q \supseteq P$ implies that $\ell \in C(P)$. Now, if g_i is the exact cumulant vector, we can (from Definition 5) substitute (4.16) into

$$g_n(P) = \sum_{\ell \in E(P)} \kappa_n(U_\ell), \qquad \forall P \subseteq P_m \tag{4.17}$$

obtaining

$$\sum_{Q \supseteq P} g_i(Q) = \sum_{Q \supseteq P} \sum_{\ell \in E(Q)} \kappa_i(U_\ell) = \sum_{\ell \in C(P)} \kappa_i(U_\ell) = f_i(P)$$

The last step follows from Lemma 4.4 (ii). Hence (i) $\implies$ (ii).

To prove that (ii) $\implies$ (i), suppose that f_i and g_i satisfy (4.10). By (4.16),

$$\sum_{Q \supseteq P} g_i(Q) = \sum_{Q \supseteq P} \sum_{\ell \in E(Q)} \kappa_i(U_\ell) \tag{4.18}$$

for all $P \subseteq P_m$. We will use (4.18) to show that g_i satisfies (4.17) by strong induction over $|P|$. In the $|P| = n$ base case, the only possible set is $P = P_m$, for which (4.18) reduces to $g_i(P_m) = \sum_{\ell \in E(P_m)} \kappa_i(U_\ell)$. Now suppose that (4.17) holds for all P with $|P| \geq i$ for some $j \in [2, n]$. Let $P \subseteq P_m$ such that $|P| = j - 1$, and observe that

$$\sum_{Q \supseteq P} g_i(Q) = g_i(P) + \sum_{Q \supset P} \sum_{\ell \in E(Q)} \kappa_i(U_\ell)$$

by the inductive hypothesis. Substituting this equation in to (4.18) and simplifying, we obtain (4.17). Hence (4.17) holds for all $P \subseteq P_m$, so (ii) $\implies$ (i).

To prove the final statement, note that the inversion stage of Algorithm 1 defines the map g_n according to (4.11), where f_n is the common cumulant vector (per Lemma 4.4 (iii)), by line 10. It follows from the equivalence proven in this lemma that g_n is the exact cumulant vector.

4.8.3 Proof of Lemma 4.7

If $g_n(P) \neq 0$, it is clear from (4.17) that $E(P)$ is non-empty, which implies that some column of the routing matrix $\mathbf{R}_\ell$ satisfies $\chi(P, P_m) = \mathbf{R}_\ell$. Now suppose that Assumptions 4.1 and 4.2 are true. By Assumption 4.1, the set $E(P)$ is either empty or contains a single element. By Assumption 4.2, if $E(P)$ contains a single element ℓ, it must satisfy $\kappa_n(U_\ell) \neq 0$. Therefore, if $g_n(P) = 0$, under these two assumptions, it follows that $E(P)$ is empty. Hence $\chi(P, P_m)$ is not a column of the routing matrix.

Per Lemma 4.6, the vector g_n in Algorithm 4.1 is the exact cumulant vector by line 10, so we can apply the above result to g_n in the reconstruction stage of the algorithm, yielding statement (iv) of Theorem 4.1.

4.8.4 Proof of Theorem 4.8

There are at most $\binom{n}{i} = O(n^i)$ size-i sets, so $\texttt{Nonzero}(f_i(P))$ is evaluated $O(n^i)$ times to compute $\mathcal{P}$. The worst-case runtime occurs when $|\{P \in \mathcal{P} : P \subseteq B\}| < t(|B|, i)$ for each iteration of the while loop, in which case the variable B takes on the value of every subset (with size at least i) of every original set in $\mathcal{B}$ precisely once (because the collection $\mathcal{X}$ tracks which sets have already been processed, preventing redundant iterations of the while loop). Thus, there are $O(2^q)$ iterations of the while loop.

To prove (ii), observe that every set added to $\mathcal{B}'$ was originally in the queue $\mathcal{B}$, and that sets in the queue are either from the original collection $\mathcal{B}$, or they are subsets of a previous element in the queue. Hence every set in $\mathcal{B}'$ is a subset of a set in the original $\mathcal{B}$, so the support estimate of $\mathcal{B}'$ is a subset of the original support estimate. To prove (iii), suppose that P is in the support estimate of $\mathcal{B}'$, so that some $B' \in \mathcal{B}'$ contains

P. Sets are only added to $\mathcal{B}'$ on line 5, and the set must satisfy either $|B'| < i$ or $|\{P' \in \mathcal{P} : P' \subseteq B'\}| \geq t(|B'|, i)$.

4.8.5 Proof of Lemma 4.9

Let P be in the support estimate of $\mathcal{B}$ with $|P| \leq s$. We can split the Möbius inversion formula into two parts:

$$\begin{aligned} g_i(P) = &\sum_{Q \supseteq P : |Q| \leq s} (-1)^{|Q|-|P|} f_i(Q) \\ &\quad + \sum_{R \supseteq P : |R| > s} (-1)^{|R|-|P|} f_i(R) \end{aligned}$$

Focus on the second sum, and let $R \supseteq P$ such that $|R| > s$. Condition (iii) implies that

$$f_i(R) = \sum_{Q \supseteq R} g_i(Q) = \sum_{B \in \mathcal{B} : B \supseteq R} g_i(B)$$

so we can simplify the second sum by

$$\begin{aligned} &\sum_{R \supseteq P : |R| > s} (-1)^{|R|-|P|} f_i(R) \\ &\quad = \sum_{R \supseteq P : |R| > s} (-1)^{|R|-|P|} \sum_{B \in \mathcal{B} : B \supseteq R} g_i(B) \\ &\quad = (-1)^{-|P|} \sum_{B \in \mathcal{B}} g_i(B) \sum_{B \supseteq R \supseteq P : |R| > s} (-1)^{|R|} \\ &\quad = (-1)^{-|P|} \sum_{B \in \mathcal{B}} g_i(B) \sum_{j=s+1}^{|B|} (-1)^j \binom{|B| - |P|}{j - |P|} \\ &\quad = \sum_{B \in \mathcal{B}} (-1)^{s+1-|P|} \binom{|B| - |P| - 1}{s - |P|} g_i(B) \end{aligned}$$

Finally, observe that $g_i(B) = f_i(B)$, since there are no proper supsersets of B in the support estimate of $\mathcal{B}$ (due to condition (i)).

Chapter 5

Network Flow Estimation

This chapter was first published in the *36th Conference on Neural Information Processing Systems* [121].

Flow networks are ubiquitous in natural and engineered systems, and in order to understand and manage these networks, one must quantify the flow of commodities across their edges. This chapter considers the estimation problem of predicting unlabeled edge flows from nodal supply and demand. We propose an implicit neural network layer that incorporates two fundamental physical laws: conservation of mass, and the existence of a constitutive relationship between edge flows and nodal states (e.g., Ohm's law). Computing the edge flows from these two laws is a nonlinear inverse problem, which our layer solves efficiently with a specialized contraction mapping. Using implicit differentiation to compute the solution's gradients, our model is able to learn the constitutive relationship within a semi-supervised framework. We demonstrate that our approach can accurately predict edge flows in AC power networks and water distribution systems.

5.1 Introduction

Network flows are a fundamental aspect of modern society, from traffic and communication networks to power and water distribution systems. Many critical infrastructures are well-modeled as graphs, with edges that transport vital commodities [106]. Beyond infrastructure, network flows are also central to models in epidemiology, ecology, medicine, and chemical networks. Their dynamics have been well-studied in compartmental systems theory [66]. Given the prevalence of flow networks in natural and engineered systems, predicting flows in these networks is an important learning task that may facilitate monitoring, control, optimization, and protection of these networks.

While domain-specific tools to predict network flows have been around for a while, the machine learning community has only recently taken an interest in general-purpose models for network flows. [71] predicts edge flows from partial measurements by making a smoothing assumption, i.e., by minimizing nodal flow divergence. [111] improves on this approach by adding a trainable regularizer that can incorporate side information. Both of these approaches are centered on a notion of approximate conservation, i.e., that the net inflow to each node should be near zero. Since conservation of mass is a universal constraint on network flows, imposing this conservation law is an important step toward embedding physics into the model.

But the conservation law alone is not enough to uniquely determine flow, which is why both [71] and [111] rely on heuristic regularizers to select the "best" conservation-respecting flow. In fact, physical networks are often governed by a *pair* of physical laws: the conservation law, and a *constitutive relationship*, which specifies the magnitude and direction of each edge flow based on "effort" variables at each incident node (e.g., pressure or voltage). For example, in DC circuits, currents are conserved according to Kirchoff's current law, and Ohm's law is the constitutive relationship that relates current flows to nodal potentials. The conservation law and the constitutive relationship together define the unique edge flows (and nodal efforts).

5.1.1 Contributions

This chapter proposes a model for network flows that embeds both the conservation law and existence of a constitutive relationship. Our model, which we call an *Implicit Flow Network* (IFN), predicts each edge flow using a trainable nonlinear function of latent nodal variables. These latent variables are constrained to a manifold wherein the conservation law is satisfied. In addition to introducing IFN, we offer the following contributions: (i) a contraction algorithm that is able to both evaluate the IFN layer and backpropagate gradients through it, (ii) an explicit upper bound on the number of iterations required by this algorithm, (iii) a rigorous theoretical comparison between IFN and the state-of-the-art flow estimation methods in [71, 111], and (iv) numerical experiments from several AC power networks and water distribution systems that indicate IFN can significantly outperform these baselines on the flow estimation task. Additionally, because IFN requires a nonlinearity with a constrained slope, we provide (v) a novel "derivative-constrained perceptron", which is essentially a trainable activation function with upper and lower bounds on its slope.

5.1.2 Related Work

Network Flow Estimation Flows on graphs are a classical topic in computer science [46], and flow forecasting has long been studied in specific domains like traffic [89], but interest in the flow estimation task from a machine learning perspective appears to be relatively recent. Deep learning algorithms have been used to predict traffic flows [88, 138] and power flows [15], but [71] and [111] appear to be the first papers to propose methods for domain-agnostic flow prediction, based on the notion of divergence minimization.

Implicit Neural Networks IFN belongs to a growing class of models called implicit neural networks, which do not explicitly state the output of the model; rather, they describe a desired relationship between the model's inputs and outputs. In the prevailing implicit framework, the output is defined as a fixed point of a trainable perceptron. This approach was introduced in [7] as a "deep equilibrium network". Subsequent work has developed new frameworks for ensuring the existence of the fixed point and computing it [134, 105, 44, 67, 47]. Other types of implicit neural networks include neural ODEs [24] and layers that solve convex optimization problems [1] and Nash equilibria [61].

Graph Neural Networks Graph neural networks (GNN) are a diverse family of models for network-related learning tasks that incorporate graph structure directly into the model. GNNs can typically be classified into three types, in increasing order of generality [19, §5.3]: convolutional models [77, 59], attentional models [130], and message-passing models [48, 8]. Recently, [56] proposed an implicit graph convolutional network. Analogously, IFN can be interpreted as an implicit message-passing GNN, with flows serving as messages and latent nodal variables acting as an embedding.

5.1.3 Preliminaries and Notation

Given a directed graph $G = (\mathcal{V}, \mathcal{E})$, the signed incidence matrix $B \in \{-1, 0, 1\}^{|\mathcal{V}| \times |\mathcal{E}|}$ is the matrix with entries

$$B_{i,e} = \begin{cases} 1, & i \text{ is the head of } e \\ -1, & i \text{ is the tail of } e \\ 0, & \text{else} \end{cases}, \quad \forall i \in \mathcal{V} \text{ and } e \in \mathcal{E}$$

Flow	Nodal Variable	$h(y) =$
DC Current	Voltage	y
DC Power	Voltage	y^2
AC Power (lossless)	Voltage Angle	$\sin(y)$
Water Flow Rate	Hydraulic Head	$\operatorname{sgn}(y)\|y\|^{0.54}$
Mechanical Force Networks	Position	y

Table 5.1: Examples of physical flow networks and their constitutive relationships.

For an undirected graph, the signed incidence matrix is obtained by assigning an aribtrary orientation to each edge. For each $i \in \mathcal{V}$, let $\mathcal{N}_{\text{in}}(i), \mathcal{N}_{\text{out}}(i) \subset \mathcal{V}$ be the in-neighbors and out-neighbors of i.

Given a vector $x \in \mathbb{R}^n$, we use the notation $[x]$ to denote the diagonal matrix $\operatorname{diag}(x) \in \mathbb{R}^{n \times n}$. Where such notation would be unclear (e.g., may be confused with brackets to indicate order of operations), we fall back on the $\operatorname{diag}(\cdot)$ notation. We write $x^\perp$ to refer to the vector space that is orthogonal to x, i.e., the space $\{x' \in \mathbb{R}^n : x^\mathsf{T} x' = 0\}$. Given a positive definite diagonal matrix $D \in \mathbb{R}^{n \times n}$, we write $||x||_{2,D}$ to represent the weighted 2-norm $\|D^{\frac{1}{2}} x\|_2$. Given any matrix M, M_i is the ith column vector of M, and $M^{(j)}$ is the transpose of the jth row vector.

5.2 Implicit Flow Networks

IFN is inspired by the physics of network systems. In many physical networks, nodes "communicate" through the exchange of a commodity, like power, water, or force, which can be represented as edge flows. Flows obey a conservation law: for all $i \in \mathcal{V}$,

$$0 = \overbrace{u_i + \sum_{j \in \mathcal{N}_{\text{in}}(i)} f_{(i,j)}}^{\text{net inflow}} - \overbrace{\sum_{j' \in \mathcal{N}_{\text{out}}(i)} f_{(i,j')}}^{\text{net outflow}}, \tag{5.1}$$

where $u \in \mathbb{R}^{|\mathcal{V}|}$ are nodal inflows from outside the network, and $f \in \mathbb{R}^{|\mathcal{E}|}$ are the edge flows. Furthermore, the flows are related to nodal variables through a constitutive relationship (CR); there is some strictly increasing function h such that, for all $(i, j) \in \mathcal{E}$,

$$f_{(i,j)} = a_{(i,j)} h(x_i - x_j), \tag{5.2}$$

where $a \in \mathbb{R}^{|\mathcal{E}|}$ are edge weights and $x \in \mathbb{R}^{|\mathcal{V}|}$ are nodal “efforts” or “potentials.” For example, in DC power networks, the CR is Ohm’s law $f_{(i,j)} = r^{-1}_{(i,j)}(x_i - x_j)$, where r are resistances and x are voltages. In lossless AC networks, the CR is the active power flow equation $f_{(i,j)} = a_{(i,j)} \sin(x_i - x_j)$, where the edge weights are a function of line parameters and x are voltage angles [83, §6.4]. In water distribution systems, the CR is the Hazen-Williams formula [45, Sec. 8.15]. Table 5.1 lists several flow networks, the physical interpretation of the effort variables x, and the flow function h.

We propose IFN as a layer that predicts edge flows based on these two physical laws—conservation and the existence of a CR:

Definition 5.1 (Implicit Flow Network). *An implicit flow network (IFN) is a module with the following components:*

(i) fixed parameters $0 < d_{\min} \leq d_{\max}$*,*

(ii) trainable parameters $\theta \in \mathbb{R}^r$ *for some* r*, and*

(iii) a family of differentiable functions $h_\theta : \mathbb{R} \to \mathbb{R}$ *such that* $d_{\min} \leq h'_\theta(y) \leq d_{\max}$ *for all* $y \in \mathbb{R}$ *and* $\theta \in \mathbb{R}^r$*, which we call flow functions.*

The module requires each of the following inputs:

(i) a weighted, connected, undirected graph $G = (\mathcal{V}, \mathcal{E}, a)$ *with edge weights* $a \in \mathbb{R}^{|\mathcal{E}|}_{>0}$*, and*

(ii) a supply / demand vector $u \in \mathbb{R}^{|\mathcal{V}|}$ *such that* $\sum_{i \in \mathcal{V}} u_i = 0$.

The module outputs the unique vector $f \in \mathbb{R}^{|\mathcal{E}|}$ *for which there exists* $x \in \mathbb{R}^{|\mathcal{V}|}$ *such that*

$$Bf = u \tag{5.3}$$

$$f = [a] h_\theta(B^\mathsf{T} x) \tag{5.4}$$

where $B \in \{-1, 0, 1\}^{|\mathcal{V}| \times |\mathcal{E}|}$ *is the signed incidence matrix of* G*, and* h_θ *is applied element-wise. We use the notation* $\mathrm{FN}_{h,\theta}(G, u)$ *to represent the solution* f *given inputs* G *and* u*, flow functions* h*, and parameters* θ*.*

We will prove that IFNs are well-posed in Theorem 5.2. Note that (5.3) and (5.4) are just vectorized statements of the conservation law (5.1) and the CR (5.2), so these two physical laws directly define the output. The IFN's only trainable component is its flow function, parameterized by θ. In practice, we will only make calls to the *inverse* of the flow function when evaluating and backpropagating through IFN layers, so it is convenient to learn the inverse flow function directly.

We emphasize that IFNs are layers that can be situated in more complex architectures, with other models upstream estimating the supply / demand vector, edge weights, or even the topology. For example, in power systems, demand forecasting is a very well-studied problem [113, 39], and one can solve the economic dispatch problem to forecast power generation at each node [122], collectively leading to an estimate of the supply / demand vector.

5.2.1 Evaluating the Implicit Flow Network

Our approach to evaluating the implicit flow network is adapted from [69] and is illustrated in Figure 5.1. Any undirected graph G induces a direct decomposition of

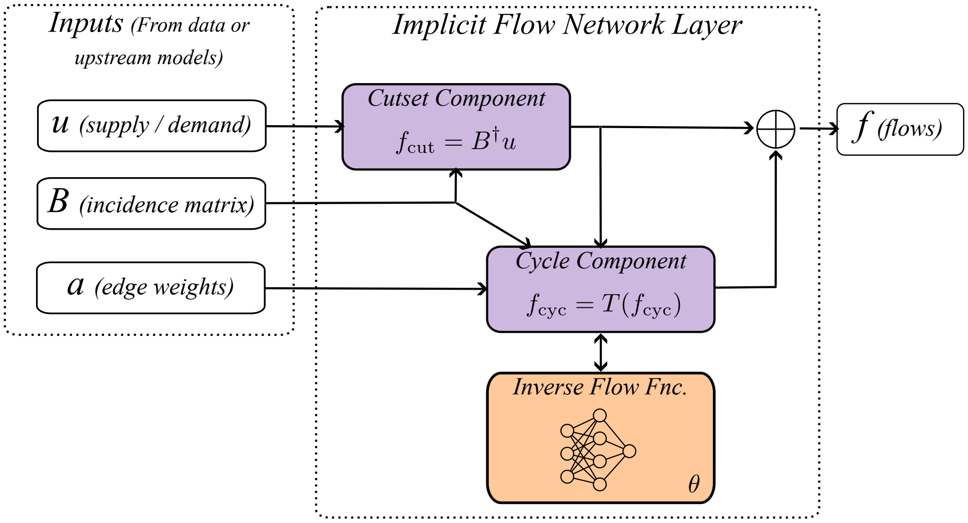
Inputs (From data or upstream models)
u (supply / demand)
B (incidence matrix)
a (edge weights)
Implicit Flow Network Layer
Cutset Component
$f_{\text{cut}} = B^{\dagger}u$
Cycle Component
$f_{\text{cyc}} = T(f_{\text{cyc}})$
Inverse Flow Fnc.
θ
f (flows)

Figure 5.1: Diagram of the IFN. Inputs are the supply / demand vector u, incidence matrix B, and edges weights a, which are either known or output from upstream models. The IFN layer separately computes the cutset component and cycle component of the flows, with a trainable model for the inverse of the flow function in the CR. These components are summed and output as the flow prediction, for downstream use.

the edge flow space $\mathbb{R}^{|\mathcal{E}|}$: given the incidence matrix $B \in \{-1, 0, 1\}^{|\mathcal{V}|\times|\mathcal{E}|}$, the *cycle space* $\ker(B)$ and *cutset space* $\operatorname{Img}(B^{\mathsf{T}})$ are orthogonal, and $\mathbb{R}^{|\mathcal{E}|} = \ker(B) \oplus \operatorname{Img}(B^{\mathsf{T}})$. We refer the reader to [22, §9.4] for a primer on cycle and cutset spaces. Accordingly, we decompose the vector $f = \mathrm{FN}_{h,\theta}(G, u)$ as $f = f_{\text{cyc}} + f_{\text{cut}}$, where $f_{\text{cyc}} \in \ker(B)$ and $f_{\text{cut}} \in \operatorname{Img}(B^{\mathsf{T}})$. The cutset component is readily determined from (5.3), since $Bf = Bf_{\text{cut}} = u$ implies that $f_{\text{cut}} = B^{\dagger}u$. Then we must analyze (5.4) to solve for f_{cyc}. Define a *cycle projection matrix* $P \in \mathbb{R}^{m\times m}$ as the oblique projection onto $\ker(B)$ parallel to $\operatorname{Img}([a]B^{\mathsf{T}})$:

$$P = I_m - [a]B^{\mathsf{T}} \left(B[a]B^{\mathsf{T}}\right)^{\dagger} B \tag{5.5}$$

Based on this projection, we define a map $T : \ker(B) \to \ker(B)$ for all $f_{\text{cyc}} \in \ker(B)$ by

$$T(f_{\text{cyc}}) = P\left(f_{\text{cyc}} - d_{\min}[a]h_{\theta}^{-1}([a]^{-1}f_{\text{cyc}} + [a]^{-1}B^{\dagger}u)\right) \tag{5.6}$$

We can show that f_{cyc} is the unique fixed point of T, and that T is a contraction mapping, leading to a simple algorithm to compute this fixed point.

Theorem 5.2 (Properties of T)**.** *Consider an implicit flow network with parameters $d_{\min}$, $d_{\max}$, and θ, with flow functions h_θ. Suppose that the inputs $G = (\mathcal{V}, \mathcal{E}, a)$ and $u \in \mathbb{1}_{|\mathcal{V}|}^{\perp}$ are given, and let $B \in \{-1, 0, 1\}^{|\mathcal{V}| \times |\mathcal{E}|}$ be the signed incidence matrix of G. The following are true:*

(i) T is a contraction mapping with respect to $||\cdot||_{2,[a]^{-1}}$, with Lipschitz constant

$$\mathrm{Lip}(T) \leq 1 - \frac{d_{\min}}{d_{\max}},$$

(ii) the sequence of iterates $f_{\text{cyc}}^{(k+1)} = T(f_{\text{cyc}}^{(k)})$ starting from any initial condition $f_{\text{cyc}}^{(0)} \in \ker(B)$ converges to a unique fixed point f_{cyc},

(iii) the output of the implicit flow network is unique and given by

$$\mathrm{FN}_{h,\theta}(G, u) = f_{\text{cyc}} + B^{\dagger} u \tag{5.7}$$

Consequently, IFN is well-posed.

Theorem 5.2 provides a simple algorithm for computing the IFN output f: pick any $f_{\text{cyc}}^{(0)} \in \ker(B)$, repeatedly apply the map T until approximate convergence, then add $B^{\dagger} u$. Some care is required when implementing this map. Since P is a dense matrix with $|\mathcal{E}|^2$ entries, it is undesirable to explicitly construct the cycle space projection matrix for large networks. Instead, in order to project a vector $v \in \mathbb{R}^{|\mathcal{E}|}$, we can use the fact that

$$w \triangleq (B[a]B^{\mathsf{T}})^{\dagger} B v = \underset{w \in \mathbb{R}^n}{\operatorname{argmin}} \left\{ ||B[a]B^{\mathsf{T}} w - Bv||_2 \right\}$$

so the projection is evaluated as $Pv = v - [a]B^\mathsf{T}w$. Using this method of projection to implement T, the fixed point iteration to compute $\text{FN}_{h,\theta}(G, u)$ is stated in Algorithm 5.1.

Algorithm 5.1 Evaluating the implicit flow network.

```
 1: $B \leftarrow$ signed incidence matrix of $G$
 2: $f_{\text{cut}} \leftarrow \text{argmin}_{f_{\text{cut}} \in \mathbb{R}^m} \{||Bf_{\text{cut}} - u||_2\}$
 3: $f_{\text{cyc}} \leftarrow \mathbb{0}_m$
 4: $\Delta f_{\text{cyc}} \leftarrow \infty \mathbb{1}_m$
 5: while $||\Delta f_{\text{cyc}}||_{2,[a]^{-1}} > \epsilon$ do
 6:     $v \leftarrow d_{\text{min}}[a]h_\theta^{-1}\left([a]^{-1}f_{\text{cyc}} + [a]^{-1}f_{\text{cut}}\right)$
 7:     $w \leftarrow \text{argmin}_{w \in \mathbb{R}^n} \left\{||B[a]B^\mathsf{T}w - Bv||_2\right\}$
 8:     $\Delta f_{\text{cyc}} \leftarrow v - [a]B^\mathsf{T}w$
 9:     $f_{\text{cyc}} \leftarrow f_{\text{cyc}} - \Delta f_{\text{cyc}}$
10: end while
11: $f \leftarrow f_{\text{cyc}} + f_{\text{cut}}$
12: return $f$
```

Theorem 5.3 (Implicit Flow Networks, Forward Pass)**.** *Consider an implicit flow network with parameters d_{min}, d_{max}, and θ, with flow functions h_θ. Suppose that the inputs $G = (\mathcal{V}, \mathcal{E}, a)$ and $u \in \mathbb{1}_{|\mathcal{V}|}^\perp$ are given, and let $B \in \{-1, 0, 1\}^{|\mathcal{V}| \times |\mathcal{E}|}$ be the signed incidence matrix of G. The following are true of Algorithm 5.1, with a tolerance of $\epsilon > 0$:*

(i) for each iteration $k = 1, 2, \ldots$ of the loop, let $f_{\text{cyc}}^{(k)}$ represent the new value of f_{cyc} defined on line 9; and let $f_{\text{cyc}}^{(0)} = \mathbb{0}_m$. Then

$$f_{\text{cyc}}^{(k+1)} = T(f_{\text{cyc}}^{(k)}), \quad \forall k \geq 0;$$

(ii) the algorithm converges with at most k^* iterations of the while loop, where

$$k^* = 1 + \frac{\log\left(d_{\text{min}}^{-1}\rho^{-1}\epsilon\right)}{\log\left(1 - \frac{d_{\text{min}}}{d_{\text{max}}}\right)} \tag{5.8}$$

and $\rho = ||[a]^{\frac{1}{2}}h_\theta^{-1}([a]^{-1}B^\dagger u)||_2$; and *(iii) the algorithm returns $f \in \mathbb{R}^{|\mathcal{E}|}$, where*

$$||f - \mathrm{FN}_{h,\theta}(G, u)||_{2,[a]^{-1}} \leq \left(\frac{d_{\max} - d_{\min}}{d_{\min}} \right) \epsilon \tag{5.9}$$

If evaluating h_θ^{-1} is sufficiently simple, then the most expensive step in the iteration is solving the ordinary least squares problem on line 7. Using a general-purpose solver, the complexity of this operation is roughly $O(|\mathcal{V}|^3)$. But $B[a]B^\top$ is a sparse Laplacian matrix, so we can use a specialized Laplacian solver that reduces the complexity to $O(|\mathcal{E}| \log^k |\mathcal{E}|)$ for some constant k [131].

The bound on the number of iterations k^* can be computed before any forward pass, since evaluating h_θ^{-1} does not require solving the IFN equations. But we can further simplify the bound by approximating $h_\theta^{-1}(0) = 0$, which is often justified because physical flow functions generally have a root at the origin. Using the fact that $(h_\theta^{-1})'(y) \leq d_{\min}^{-1} y$, we can then eliminate the dependence on h_θ^{-1}:

$$k^* \leq 1 + \log\left(1 - \frac{d_{\min}}{d_{\max}}\right) \left(\log \epsilon - \log\left(||[a]^{-\frac{1}{2}} B^\dagger u||_2\right)\right)$$

5.2.2 Computing the Gradients

In order to train the flow function and any upstream models, it is necessary to backpropagate gradients through the IFN layer. We can perform this backward pass using implicit differentiation, and it turns out that the gradients of $\mathrm{FN}_{h,\theta}(G, u)$ with respect to the parameters θ, a, and u can also be computed using Algorithm 5.1, i.e., by writing the gradient as the output of an auxiliary implicit flow network.

Theorem 5.4 (Gradients)**.** *Consider an implicit flow network with parameters $d_{\min}$, $d_{\max}$, and θ, with flow functions h_θ. Suppose that the inputs $G = (\mathcal{V}, \mathcal{E}, a)$ and $u \in \mathbb{1}_{|\mathcal{V}|}^{\perp}$*

are given, and let $B \in \{-1, 0, 1\}^{|\mathcal{V}| \times |\mathcal{E}|}$ *be the signed incidence matrix of* G*. Let* $f = \mathrm{FN}_{h,\theta}(G, u)$*, and let* w *be a scalar entry of* θ*,* a*, or* u*. We can compute the derivatives* $\frac{df}{dw}$ *as follows.*

Define a vector of flow functions $g : \mathbb{R}^{|\mathcal{E}|} \to \mathbb{R}^{|\mathcal{E}|}$ *by*

$$g(\eta) = \mathcal{D}^{-1}\left(\eta - [a]^{-1}\frac{\partial v}{\partial w}\right), \quad \forall \eta \in \mathbb{R}^{|\mathcal{E}|} \tag{5.10}$$

where $\mathcal{D} \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{E}|}$ *is the diagonal matrix with entries*

$$\mathcal{D}_{ee} = \left.\frac{dh_\theta^{-1}(y_e)}{dy_e}\right|_{y_e = a_e^{-1} f_e}, \quad \forall e \in \mathcal{E} \tag{5.11}$$

and $v = [a]h_\theta^{-1}([a]^{-1}f_{\mathrm{cyc}} + [a]^{-1}B^\dagger u)$*. Then*

$$\frac{df}{dw} = \mathrm{FN}_{g,\cdot}(G, \mathbb{0}_n) + B^\dagger \frac{du}{dw} \tag{5.12}$$

(We use the notation $\cdot$ *in place of* θ*, since* g *has no trainable parameters.) Furthermore, the derivative constraint parameters* $d_{\min}, d_{\max}$ *from the original implicit flow network are valid for the new implicit flow network.*

In other words, to compute the gradient with respect to a parameter, we perform a single evaluation of the implicit flow network. In order to compute the derivatives with respect to some parameter or input w, we first evaluate the partial derivatives $\frac{\partial v}{\partial w}$ and the total derivatives $\frac{du}{dw}$. Then we construct the flow functions g according to (5.10), and solve an implicit flow network to find $\frac{df}{dw}$ according to (5.12). It is easy to evaluate $\frac{du}{dw}$, but for convenience, we provide the values of $\frac{\partial v}{\partial w}$ below:

$$\frac{\partial v}{\partial \theta_i} = [a]\frac{dh_\theta^{-1}([a]^{-1}f)}{d\theta_i}, \qquad \frac{\partial v}{\partial a_e} = \mathrm{diag}\left(h_\theta^{-1}([a]^{-1}f) - [a]^{-1}\mathcal{D}f\right)_e, \qquad \frac{\partial v}{\partial u_i} = \left(\mathcal{D}B^\dagger\right)_i$$

5.3 Comparison with Optimization Models

Both of the state-of-the-art methods for flow estimation, from [71] and [111], use an optimization problem to predict flows. After a suitable transformation to incorporate external flow injections u, we can state this optimization problem as

$$\hat{f} = \underset{f \in \mathbb{R}^{|\mathcal{E}|}}{\operatorname{argmin}} \left\{ ||f||_{2,[q]}^2 + \lambda^2 ||Bf - u||_2^2 \text{ s.t. } f_e = \tilde{f}_e, \;\; \forall \text{ labeled edges } e \in \mathcal{E} \right\} \tag{5.13}$$

where $\lambda > 0$, and $q > \mathbb{0}_m$ is some vector of edge weights. In [71], $q = \mathbb{1}_m$, while [111] allows q to be the output of a neural network. IFN is not explicitly an optimization problem, but it can be cast as one that is similar to (5.13):

Theorem 5.5 (Optimization Form of IFN)**.** *Consider an IFN with flow function h_θ. Suppose that the inputs $G = (\mathcal{V}, \mathcal{E}, a)$ and $u \in \mathbb{1}_{|\mathcal{V}|}^{\perp}$ are given, and let $B \in \{-1, 0, 1\}^{|\mathcal{V}| \times |\mathcal{E}|}$ be the signed incidence matrix of G. Then the IFN output can be stated as the solution of a convex optimization problem:*

$$\mathrm{FN}_{h,\theta}(G, u) = \underset{f \in \mathbb{R}^m}{\operatorname{argmin}} \left\{ \sum_{e \in \mathcal{E}} \int_0^{f_e} h_\theta^{-1}(a_e^{-1} z) \; dz \;\; s.t. \;\; Bf = u \right\} \tag{5.14}$$

Theorem 5.5 can be interpreted as a nonlinear generalization of the Thomson principle from electrical circuits theory [41]. Interestingly, the theorem sets up a direct comparison between IFN and the models in [71] and [111]. If the flow function h_θ is the identity map, then (5.14) can be simplified as

$$\mathrm{FN}_{h,\theta}(G, u) = \underset{f \in \mathbb{R}^m}{\operatorname{argmin}} \left\{ \sum_{e \in \mathcal{E}} ||f||_{2,[a]^{-1}}^2 \text{ s.t. } Bf = u \right\} \tag{5.15}$$

Ignoring the constraints from labeled flows, we can interpret (5.13) as using a penalty method to approximate the output of an IFN with a linear flow function. Thus, we have three distinct differences between IFN and the optimization-based approaches. First, IFN allows for a nonlinear flow function, while [71] and [111] implicitly assume a linear CR. Second, IFN imposes flow conservation as a hard constraint rather than an approximate constraint (which is a limitation if u is uncertain). Finally, IFN does not incorporate flow measurements directly; rather, the model exploits these measurements during training to learn the proper flow function (and train any upstream models for the IFN inputs), making it less sensitive to noise in the labeled flows.

5.4 Models for Flow Functions

In order to implement an IFN, it is necessary to parameterize its inverse flow function h_θ^{-1}. Since the flow function is essentially a trainable activation function, i.e., a scalar nonlinearity, simple models are likely to be sufficient. The main difficulty with selecting a flow function is that its slope must be bounded by $d_{\min} \leq h'_\theta(y) \leq d_{\max}$ for all $y \in \mathbb{R}$. This section proposes a simple scalar nonlinearity that is guaranteed to respect arbitrary upper and lower bounds on its slope.

Definition 5.6 (Derivative-Constrained Perceptron)**.** *Let $k \in \mathbb{Z}_{>0}$ be a hidden layer size, let $a, b, c \in \mathbb{R}^k$ be freely trainable parameters (encoded within the parameter vector θ), and let σ be a non-expansive activation. Let $p, q \geq 1$ such that $p^{-1} + q^{-1} = 1$, and let $\bar{d}_{\min} \leq \bar{d}_{\max} \in \mathbb{R}$. Then the derivative-constrained perceptron $N(x, \theta)$ is the 3-layer neural network defined by*

$$\bar{c}(\theta) = \left(1 - \frac{(||c||_p||a||_q - 1)_+}{||c||_p||a||_q}\right) c \tag{L1}$$

$$N_0(x, \theta) = \bar{c}^\mathsf{T}(\theta)\sigma(ax + b) \tag{L2}$$

$$N(x, \theta) = \left(\frac{\bar{d}_{\max} - \bar{d}_{\min}}{2}\right) N_0(x, \theta) + \left(\frac{\bar{d}_{\max} + \bar{d}_{\min}}{2}\right) x \tag{L3}$$

Intuitively, (L1) re-scales c so that the perceptron in (L2) is guaranteed to be non-expansive in x. Then (L3) re-centers and re-scales the derivatives of the perceptron from the range $[-1, 1]$ to $[\bar{d}_{\min}, \bar{d}_{\max}]$.

Theorem 5.7 (Derivative-Constrained Perceptron)**.** *Let* $\mathrm{N}(x, \theta)$ *be a derivative-constrained perceptron with* $\bar{d}_{\min} \leq \bar{d}_{\max} \in \mathbb{R}$*. Then for all parameter values* θ*,*

$$\bar{d}_{\min} \leq \frac{d}{dx} \mathrm{N}(x, \theta) \leq \bar{d}_{\max}, \quad \forall x \in \mathbb{R} \tag{5.16}$$

Note that the values $\bar{d}_{\min}, \bar{d}_{\max}$ in Definition 5.6 and Theorem 5.7 are distinct from the IFN parameters $d_{\min}, d_{\max}$. Since we parameterize the *inverse* flow function h_θ^{-1} in IFN, one shoudl set $\bar{d}_{\min} = d_{\max}^{-1}$ and $\bar{d}_{\max} = d_{\min}^{-1}$ to implement h_θ^{-1} with a derivative-constrained perceptron.

5.5 Numerical Experiments

We studied the transductive task of predicting unlabeled flows, given that some labeled flows in the same network are known. If the edges $\mathcal{E}$ are partitioned into a labeled set $\mathcal{E}_l$ and an unlabeled set $\mathcal{E}_u$, the task is to predict the missing flows $\{f_e : e \in \mathcal{E}_u\}$ given the labeled flows $\{f_e : e \in \mathcal{E}_l\}$. For each network, we randomly selected a fraction of the edges to be labeled edges, and we trained IFN and baselines on the labeled edges. Then we evaluated the RMSE of the flows predicted for the unlabeled edges $\mathcal{E}_u$ to compute the testing error. See Section 5.8 for full details. Code is available at `https://github.com/KevinDalySmith/implicit-flow-networks`.

5.5.1 Datasets

AC Power We selected 6 standard power network test cases. The first 4 test cases (IEEE-57, IEEE-118, IEEE-145, and IEEE-300) are synthetic transmission system test cases, while the remaining cases ACTIVSg200 and ACTIVSg500 are similar to the Illinois and South Carolina power grids, respectively [11]. Each test case contains the topology and electrical parameters of the power network, as well as baseline demands and power injections at each node. While branch resistances are typically small, we set them to zero to ensure lossless transmission lines. We used the MATPOWER toolbox [144] to solve the power flow equations, then recorded the active power flows on each branch (f), computed the net active power injections at each node (u), and selected relevant electrical parameters as edge attributes (series reactance, tap ratio, and voltage magnitude at the two incident nodes).

Water Distribution We selected 3 sample water distribution networks from the ASCE Task Committee on Research Databases for Water Distribution Systems database [62], representing municipal water distribution systems in Fairfield, CA, Bellingham, WA, and Harrisburg, PA. Each network contains the topology of the distribution system, as well as the characteristics of pipes and other network elements and nodal demands. We used the WNTR package [78] to compute the flow rates through each pipe (f), net inflow rate at each node (u), and edge weights associated with each pipe.

5.5.2 Models and Experiment Details

IFN Architecture In order to use the IFN layer to predict power flows, we created a two-layer model. The first layer estimates positive edge weights $a \in \mathbb{R}^{|\mathcal{E}|}$ according to $a_e = \exp\left(\mathrm{L}(z_e)\right)$ for all $e \in \mathcal{E}$, where L is a linear module, and z_e is the log-transformed vector of edge attributes. The second layer is an IFN. To predict water flows, we used an IFN layer alone, supplying the edge weights from the dataset as input (rather than learning them from other edge attributes). For both water and power, the IFN layer uses a derivative-constrained perceptron as the inverse flow function ($k = 128$, $p = q = \frac{1}{2}$) with a ReLU activation function. For power, we set $d_{\min} = 0.4$ and $d_{\max} = 2$; and for water, $d_{\min} = 0.2$ and $d_{\max} = 20$.

Baselines We compared the IFN model against four baselines. The minimum divergence method (*Div*) from [71] minimizes the nodal divergence $||Bf||_2^2$ and a regularization term $\lambda||f||_2^2$. The bilevel optimization methods from [111] replace the uniform regularizer with a weighted regularizer $||f||^2_{2,[q]}$, where q is a vector of weights. In *Bil-MLP* and *Bil-GCN*, q is the output of either a 2-layer MLP or GCN model with edge attributes as inputs (we use 64 nodes in each hidden layer with ReLU activations). In *Bil-True*, we specify q as the reciprocal of the coefficient in the linearized CR for AC power networks, so that *Bil-True* approximates (5.15) with a as the ground-truth edge weight. For water experiments, *Bil-True* uses the same edge weights as the IFN model.

All of the baselines assume that nodal divergence Bf should be approximately zero, but nodes in power networks inject and withdraw power according to the supply / demand vector u, resulting in nonzero divergence. Thus, when we evaluate the baselines, we transform the power network into a divergence-free network by introducing a "source node", adding an edge from the source node to all nodes in $\mathcal{V}$, and treating the entries of u as the flows along each corresponding virtual edge.

5.5.3 Results

Figure 5.2 reports the results for the AC power networks, and Figure 5.3 reports the results for water distribution systems. In both types of networks, the IFN model

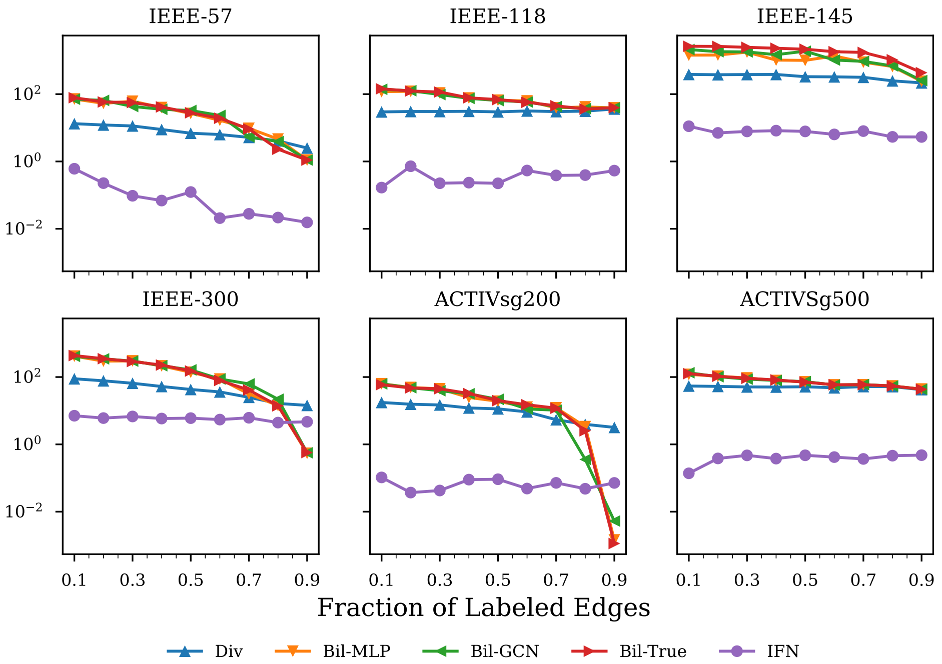

Figure 5.2: Results for missing flow prediction in AC power networks. Reported values are the RMSE (in units of MW) on the testing set, averaged across 10 trials. Note the vertical axis is in a log scale.

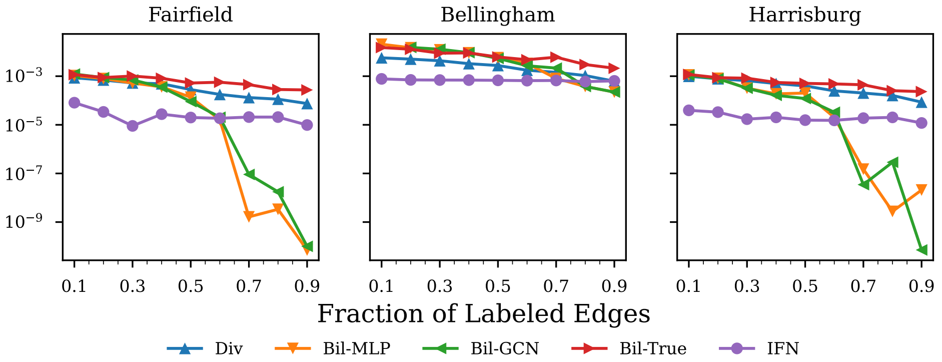

Figure 5.3: Results for missing flow prediction in water distribution systems. Reported values are the RMSE (in units of m^3/s) on the testing set, averaged across 10 trials.

significantly outperforms the baselines on all of the networks when a small fraction of edges are labeled (less than 80% in power and less than 60% in water). While the other baselines tend to improve as more labeled edges are made available for training, IFN achieves near-optimal performance with as few as 10% of the edges labeled.

5.6 Conclusion

In this chapter, we have introduced an implicit model for network flows that incorporates physics through a conservation law and through the existence of a constitutive relationship between flows and nodal variables. We have demonstrated that a simple architecture using this model can learn to accurately predict active power flows in AC networks and water distribution systems. Future work may investigate more elaborate architectures using IFN as a layer, wherein the supply / demand vector, edge weights, or even the graph itself could be predicted from upstream models, and the flows themselves used for downstream tasks. Another interesting extension may be to extend our method to networks with higher-order interactions, i.e., hypergraphs [73] and simplicial complexes [107, 136].

IFN has some limitations that should also be addressed in future work. IFN assumes that the graph is undirected, which does not adequately model networks with unidirectional flows (e.g., traffic) or lossy flows (e.g., resistive power grids). IFN also assumes a CR that depends on the *difference* between nodal variables. This form appears frequently in physical systems, but in other network flow models (like Daganzo traffic models [30]), the CR has a more general dependence on the nodal variables. These limitations may be addressed with extensions of IFN's contraction algorithm.

5.7 Proofs

5.7.1 Proof of Theorem 5.2

To prove statement (i), choose any $f_{\mathrm{cyc}} \in \ker(B)$, let $y = [a]^{-1} f_{\mathrm{cyc}} + [a]^{-1} B^{\dagger} u$ for brevity, and observe that

$$\begin{aligned}\left\|\frac{\partial T(f_{\mathrm{cyc}})}{\partial f_{\mathrm{cyc}}}\right\|_{2,[a]^{-1}} &= \left\|[a]^{-\frac{1}{2}} P\left(I_m - d_{\min}[a]\frac{\partial h_\theta^{-1}(y)}{\partial y}[a]^{-1}\right)[a]^{\frac{1}{2}}\right\|_2 \\ &\leq \|[a]^{-\frac{1}{2}} P[a]\|_2 \left\|[a]\left(I_m - d_{\min}[a]\frac{\partial h_\theta^{-1}(y)}{\partial y}[a]^{-1}\right)[a]^{\frac{1}{2}}\right\|_2 \\ &= \left\|[a]^{-\frac{1}{2}}\left(I_m - d_{\min}[a]\frac{\partial h_\theta^{-1}(y)}{\partial y}[a]^{-1}[a]^{-1}\right)[a]^{\frac{1}{2}}\right\|_2\end{aligned}$$

where $\|[a]^{-\frac{1}{2}} P[a]^{\frac{1}{2}}\|_2 = 1$ because $[a]^{-\frac{1}{2}} P[a]^{\frac{1}{2}}$ is a symmetric and idempotent matrix, i.e., an orthogonal projection. Then

$$\left\|\frac{\partial T(f_{\mathrm{cyc}})}{\partial f_{\mathrm{cyc}}}\right\|_{2,[a]^{-1}} = \max_{e\in\mathcal{E}} \left|1 - d_{\min}(h_\theta^{-1})'(y_e)\right| \leq 1 - \frac{d_{\min}}{d_{\max}}$$

Hence

$$\mathrm{Lip}(T) = \sup_{f_{\mathrm{cyc}}\in\mathbb{R}^m} \left\|\frac{\partial T(f_{\mathrm{cyc}})}{\partial f_{\mathrm{cyc}}}\right\|_{2,[a]^{-1}} \leq 1 - \frac{d_{\min}}{d_{\max}} < 1$$

Then statement (ii) follows from statement (i) and the Banach fixed point theorem. To prove statement (iii), observe that $f_{\mathrm{cyc}} = P f_{\mathrm{cyc}}$, so $f_{\mathrm{cyc}} = T(f_{\mathrm{cyc}})$ if and only if

$$P[a]h_\theta^{-1}\left([a]^{-1} f\right) = \mathbb{0}_m \tag{5.17}$$

where $f = f_{\mathrm{cyc}} + B^{\dagger} u$. But $\ker(P[a]) = \mathrm{Img}(B^{\mathsf{T}})$, so (5.17) is equivalent to the existence of $x \in \mathbb{R}^n$ such that

$$h^{-1}([a]^{-1} f) = B^{\mathsf{T}} x \tag{5.18}$$

and (5.18) is equivalent to (5.4).

5.7.2 Proof of Theorem 5.3

To prove statement (i), let $k \geq 0$ and consider iteration $k+1$ of the loop. The iteration first defines $v = d_{\min}[a]h_\theta^{-1}\left([a]^{-1}f_{\text{cyc}}^{(k)} + [a]^{-1}f_{\text{cut}}\right)$ on line 6. Then on line 7,

$$w = \operatorname*{argmin}_{w \in \mathbb{R}^n} \left\{||B[a]B^\mathsf{T} w - Bv||_2\right\} = \left(B[a]B^\mathsf{T}\right)^\dagger Bv$$

and line 8 defines

$$\Delta f_{\text{cyc}} = v - [a]B^\mathsf{T} w = \left(I_m - [a]B^\mathsf{T}\left(B[a]B^\mathsf{T}\right)^\dagger\right) v = Pv$$

Finally, on line 9,

$$f_{\text{cyc}}^{(k+1)} = f_{\text{cyc}}^{(k)} - Pv = f_{\text{cyc}}^{(k)} - d_{\min}P[a]h_\theta^{-1}\left([a]^{-1}f_{\text{cyc}}^{(k)} + [a]^{-1}f_{\text{cut}}\right)$$

A simple inductive argument shows that $f_{\text{cyc}}^{(k)} \in \ker(B)$. The base case $f_{\text{cyc}}^{(0)} = \mathbb{0}_m$ is trivial, for all $k' \geq 0$, line 9 ensures that $f_{\text{cyc}}^{(k'+1)} \in \ker(B)$ so long as $f_{\text{cyc}}^{(k')} \in \ker(B)$. Hence $f_{\text{cyc}}^{(k)} = Pf_{\text{cyc}}^{(k)}$, and we conclude that

$$f_{\text{cyc}}^{(k+1)} = P\left(f_{\text{cyc}}^{(k)} - d_{\min}[a]h_\theta^{-1}\left([a]^{-1}f_{\text{cyc}}^{(k)} + [a]^{-1}f_{\text{cut}}\right)\right) = T(f_{\text{cyc}}^{(k)})$$

To prove statement (ii), recall from Theorem 5.2 that $\text{Lip}(T) \leq 1 - d_{\max}^{-1}d_{\min}$, which (together with statement (i)) implies that, for all $k \geq 0$,

$$\begin{aligned}
||f_{\text{cyc}}^{(k+1)} - f_{\text{cyc}}^{(k)}||_{2,[a]^{-1}} &\leq \left(1 - \frac{d_{\text{min}}}{d_{\text{max}}}\right)^{k} ||f_{\text{cyc}}^{(1)} - f_{\text{cyc}}^{(0)}||_{2,[a]^{-1}} \\
&= d_{\text{min}} \left(1 - \frac{d_{\text{min}}}{d_{\text{max}}}\right)^{k} ||P[a]h_{\theta}^{-1}\left([a]^{-1}B^{\dagger}u\right)||_{2,[a]^{-1}} \\
&= d_{\text{min}} \left(1 - \frac{d_{\text{min}}}{d_{\text{max}}}\right)^{k} \rho
\end{aligned}$$

The algorithm terminates after iteration k if and only if $||f_{\text{cyc}}^{(k)} - f_{\text{cyc}}^{(k-1)}||_{2,[a]^{-\frac{1}{2}}} \leq \epsilon$, so the algorithm will have terminated after k^{*} iterations if

$$d_{\text{min}} \left(1 - \frac{d_{\text{min}}}{d_{\text{max}}}\right)^{k^{*}-1} \rho \leq \epsilon$$

which is equivalent to

$$k^{*} \geq 1 + \frac{\log\left(d_{\text{min}}^{-1}\rho^{-1}\epsilon\right)}{\log\left(1 - \frac{d_{\text{min}}}{d_{\text{max}}}\right)}$$

Finally, to prove statement (iii), note that the algorithm terminates after iteration k as soon as

$$||f_{\text{cyc}}^{(k)} - f_{\text{cyc}}^{(k-1)}||_{2,[a]^{-1}} \leq \epsilon$$

If f_{cyc} is the true fixed point of T, then using a general property of contraction mappings,

$$\begin{aligned}
||f_{\text{cyc}}^{(k)} - f_{\text{cyc}}||_{2,[a]^{-1}} &\leq \frac{\text{Lip}(T)}{1 - \text{Lip}(T)} ||f_{\text{cyc}}^{(k)} - f_{\text{cyc}}^{(k-1)}||_{2,[a]^{-1}} \\
&\leq \left(\frac{d_{\text{max}} - d_{\text{min}}}{d_{\text{min}}}\right) \epsilon
\end{aligned}$$

Therefore, the vector f returned by the algorithm satisfies

$$||f - \text{FN}_{h,\theta}(G,u)||_{2,[a]^{-1}} = ||f_{\text{cyc}}^{(k)} - f_{\text{cyc}}||_{2,[a]^{-1}} \leq \left(\frac{d_{\text{max}} - d_{\text{min}}}{d_{\text{min}}}\right) \epsilon$$

5.7.3 Proof of Theorem 5.4

Let $v = [a]h_\theta^{-1}([a]^{-1}f_{\text{cyc}} + [a]^{-1}B^\dagger u)$. From Theorem 5.2, we can write $f = f_{\text{cyc}} + B^\dagger u$, where f_{cyc} is the unique fixed point of T. Therefore $\frac{df}{dw} = \frac{df_{\text{cyc}}}{dw} + B^\dagger \frac{du}{dw}$, so the remainder of the proof is to show that $\frac{df_{\text{cyc}}}{dw} = \text{FN}_{g,\cdot}(G, \mathbb{0}_n)$.

Since $f_{\text{cyc}} = T(f_{\text{cyc}})$, and $Pf_{\text{cyc}} = f_{\text{cyc}}$, we have

$$f_{\text{cyc}} = P\left(f_{\text{cyc}} - d_{\min} v\right) = f_{\text{cyc}} - d_{\min} P v$$

so an equivalent characterization of f_{cyc} is the unique solution to the equations

$$\begin{aligned} Bf_{\text{cyc}} &= \mathbb{0}_n \\ Pv &= \mathbb{0}_m \end{aligned}$$

Since

$$\frac{dv}{dw} = \frac{\partial v}{\partial w} + \frac{\partial v}{\partial f_{\text{cyc}}}\frac{df_{\text{cyc}}}{dw} = \frac{\partial v}{\partial w} + \mathcal{D}\frac{df_{\text{cyc}}}{dw}$$

then differentiating and factoring out $[a]$, we obtain

$$\begin{aligned} B\frac{df_{\text{cyc}}}{dw} &= \mathbb{0}_n \\ P[a]\left([a]^{-1}\frac{\partial v}{\partial w} + [a]^{-1}\mathcal{D}\frac{df_{\text{cyc}}}{dw}\right) &= \mathbb{0}_m \end{aligned}$$

Since $\ker(P[a]) = \text{Img}(B^\mathsf{T})$, there exists $x \in \mathbb{R}^n$ such that

$$[a]^{-1}\frac{\partial v}{\partial w} + [a]^{-1}\mathcal{D}\frac{df_{\text{cyc}}}{dw} = B^\mathsf{T} x$$

which we can re-write as

$$\frac{df_{\text{cyc}}}{dw} = [a]\mathcal{D}^{-1}\left(B^{\mathsf{T}}x - [a]^{-1}\frac{\partial v}{\partial w}\right) = [a]g(B^{\mathsf{T}}x)$$

Hence $\frac{df_{\text{cyc}}}{dw}$ is the solution to

$$B\frac{df_{\text{cyc}}}{dw} = \mathbb{0}_n \tag{5.19}$$

$$\frac{df_{\text{cyc}}}{dw} = [a]g(B^{\mathsf{T}}x) \tag{5.20}$$

which is identical to (5.3)–(5.4) with $\frac{df_{\text{cyc}}}{dw}$ in place of f, $\mathbb{0}_n$ in place of u, and g in place of h_θ. Furthermore, g respects the same $d_{\min}, d_{\max}$ derivative constraints as h_θ, since for each $e \in \mathcal{E}$,

$$g_e'(\eta_e) = \frac{1}{\mathcal{D}_{ee}} = \left.\frac{dh_\theta(y_e)}{dy_e}\right|_{y_e=a_e^{-1}f_e} \in [d_{\min}, d_{\max}]$$

It follows that $\frac{df_{\text{cyc}}}{dw}$ is the output of the implicit flow network with flow functions g and parameters $d_{\min}, d_{\max}$, evaluated on the original graph G and nodal demands $\mathbb{0}_n$.

5.7.4 Proof of Theorem 5.5

Since h_θ^{-1} is increasing, the optimization problem in (5.14) has a convex cost function with linear constraints, so the KKT conditions are necessary and sufficient. Letting $x \in \mathbb{R}^m$ be a vector of Lagrange multipliers, the Lagrangian is

$$\mathcal{L} = \sum_{e\in\mathcal{E}} \int_0^{f_e} h_\theta^{-1}(a_e^{-1}z)\ dz - x^{\mathsf{T}}\left(Bf - u\right)$$

leading to the stationarity condition

$$\mathbb{0}_m^{\mathsf{T}} = \frac{\partial \mathcal{L}}{\partial f} = h_\theta^{-1}(f^{\mathsf{T}}[a]^{-1}) - x^{\mathsf{T}}B$$

which is equivalent to (5.4). Additionally, the primal constraint $Bf = u$ is equivalent to (5.3), so the minimizer of the optimization problem is identical to the output of the IFN.

5.7.5 Proof of Theorem 5.7

Due to (L3), it is clear that the derivative bounds (5.16) hold if and only if

$$\left|\frac{\mathrm{d}\mathrm{N}_0(x,\theta)}{\mathrm{d}x}\right| \leq 1, \qquad \forall x \in \mathbb{R} \tag{5.21}$$

For all $x, x' \in \mathbb{R}$, by Hölder's inequality,

$$\begin{aligned}|\mathrm{N}_0(x,\theta) - \mathrm{N}_0(x',\theta)| &= \left|\bar{c}^{\mathsf{T}}(\theta)\left(\sigma(ax+b) - \sigma(ax'+b)\right)\right| \\ &\leq ||\bar{c}(\theta)||_p \, ||\sigma(ax+b) - \sigma(ax'+b)||_q\end{aligned}$$

Since σ is non-expansive, its Lipschitz constant with respect to the q-norm is

$$\mathrm{Lip}(\sigma) = \sup_{\eta\in\mathbb{R}^k} \left|\left|\frac{\partial\sigma(\eta)}{\partial\eta}\right|\right|_q = \sup_{\eta_0\in\mathbb{R}} |\sigma'(\eta_0)| \leq 1$$

and thus

$$||\sigma(ax+b) - \sigma(ax'+b)||_q \leq ||a(x-x')||_q \leq ||a||_q|x-x'|$$

Furthermore, by (L1),

$$\begin{aligned}||\bar{c}(\theta)||_p||a||_q &= \left(1 - \frac{(||c||_p||a||_q - 1)_+}{||c||_p||a||_q}\right)||c||_p||a||_q \\ &= ||c||_p||a||_q - (||c||_p||a||_q - 1)_+ \\ &= \min\left\{1, ||c||_p||a||_q\right\}\end{aligned}$$

Test Case	MATPOWER Case Name	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$
IEEE-57	`case57`	57	135
IEEE-118	`case118`	118	297
IEEE-145	`case145`	145	567
IEEE-300	`case300`	300	709
ACTIVSg200	`case_ACTIVSg200`	200	445
ACTIVSg500	`case_ACTIVSg500`	500	1084

Table 5.2: MATPOWER test case details.

so that

$$|\mathrm{N}_0(x,\theta) - \mathrm{N}_0(x',\theta)| \leq \min\{1, ||c||_p||a||_q\}\,|x - x'| \leq |x - x'|$$

for all x, x'. Hence (5.21) is satisfied.

5.8 Experiment Details

5.8.1 AC Power Datasets

We created datasets from 6 AC power network test cases. Each dataset that we created represents a snapshot of an AC power network in its steady state, consisting of four components: the network topology (as an oriented, undirected graph), four attributes on each edge (voltage magnitude at the two incident nodes, series reactance, and tap ratio), the net power injection at each node, and the active power flow through each branch.

Original Data We generated our datasets using MATPOWER, an open-source toolbox for power system simulation in MATLAB [144]. The toolbox includes many standard test cases, which contain a network topology and tables of electrical and economic parameters for each bus (node), branch (edge), and generator. We selected 6 test cases, listed in Table 5.2. The raw data files for these test cases are available from the MATPOWER source[1],

[1] https://github.com/MATPOWER/matpower/tree/master/data

and details on the test case file format are contained in Appendix B of the user manual[2].

Data Generation After loading each test case into MATPOWER, we performed the following two modifications of the network parameters:

(i) We set branch resistances (column 3 in the branch data table) to zero, so that transmission lines in the system are lossless. This step was necessary because IFN is limited to undirected graphs, while lossy lines are more appropriately modeled with a pair of directed edges, since the power injected at one endpoint does not equal the power withdrawn from the other endpoint. Fortunately, branch resistances are typically small before this modification.

(ii) We replaced any negative series reactances (column 4 in the branch data table) with a positive value, chosen as the median of the positive series reactances in the same network. We performed this modification because negative series reactances results in *decreasing* constitutive relationships on the corresponding edges, whereas IFN assumes that the constitutive relationship is increasing. This modification only affected two networks: IEEE-145, in which 24 (4.2%) of the branches were assigned a series reactance of 0.2306; and IEEE-300, in which 1 (0.1%) of the branches was assigned a series reactance of 0.059.

We then computed the resulting power flows using the `runpf` function and recorded the results.

Pre-Processing Finally, we converted the results from the MATPOWER simulation into a PyTorch Geometric data object, with the following attributes:

- `edge_index`, the edge index tensor, containing the topology from the test case.

[2] https://matpower.org/docs/MATPOWER-manual.pdf

Test Case	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$
Fairfield	111	125
Bellingham	121	162
Harrisburg	261	286

Table 5.3: Water distribution network details.

- `x`, a tensor of net active power injections at each node, which has the property that $\mathbb{1}_n^\mathsf{T} x = 0$. (This tensor is identical to the supply / demand vector u.)
- `edge_attr`, a tensor of four relevant attributes for each edge: the voltage magnitudes at the two incident nodes, the series reactance, and the tap ratio.
- `f_true`, the tensor of active power flows on each edge simulated by MATPOWER.

The net active power injections at each node are computed according to

$$u_i = \mathrm{PG}_i - \mathrm{PD}_i - \mathrm{GS}_i \mathrm{VM}_i^2$$

where PG_i is active power generated at i, PD_i is active power demanded, GS_i is shunt conductance, and VM_i is the voltage magnitude.

5.8.2 Water Distribution Dataset

We created 3 datasets representing snapshots of municipal water distribution networks in their steady state, consisting of four components: the network topology (as an oriented, undirected graph), weights for each edge, the net inflow rates at each node, and the flow rate through each pipe.

Original Data Each of the datasets is based on a network from the ASCE Task Committee on Research Databases for Water Distribution Systems database [62]. Networks in this database contain a distribution network topology and tables of hydraulic parameters and operating characteristics for each node, pipe (edge), pump, reservoir, and storage tank in the network. We selected 3 networks, listed in Table 5.3 and plotted in Figure 5.4. The raw data files are available online[3].

Data Generation and Preprocessing We loaded each network INP file into WNTR and ran the WNTR simulator with a hydraulic accuracy of 10^{-8}. We then converted the results into a PyTorch Geometric data object, with the following attributes:

- `edge_index`, the edge index tensor, containing the topology from the test case.
- `x`, a tensor of net inflows at each node, which has the property that $\mathbb{1}_n^\intercal x = 0$.
- `edge_attr`, a tensor of three relevant attributes for each edge: the pipe length, pipe diameter, and pipe roughness coefficient.
- `f_true`, the tensor of flow rates through each pipe simulated by WNTR.

Edge weights are computed according to the formula

$$a_e = (0.27855) C_e D_e^{2.63} L_e^{-0.54} \tag{5.22}$$

where C_e is the roughness coefficient (unitless), D_e is the diameter (meters), and L_e is the pipe length (meters) [4].

5.8.3 Details on IFN

Our IFN implementation uses Algorithm 5.1 to compute the layer's forward pass. We set the maximum number of iterations in this algorithm to 100, with a tolerance of $\epsilon = 10^{-2}$ for power and $\epsilon = 10^{-4}$ for water. With the release of PyTorch 1.11.0, the

[3] http://www.uky.edu/WDST/index.html

[4] https://wntr.readthedocs.io/en/latest/hydraulics.html

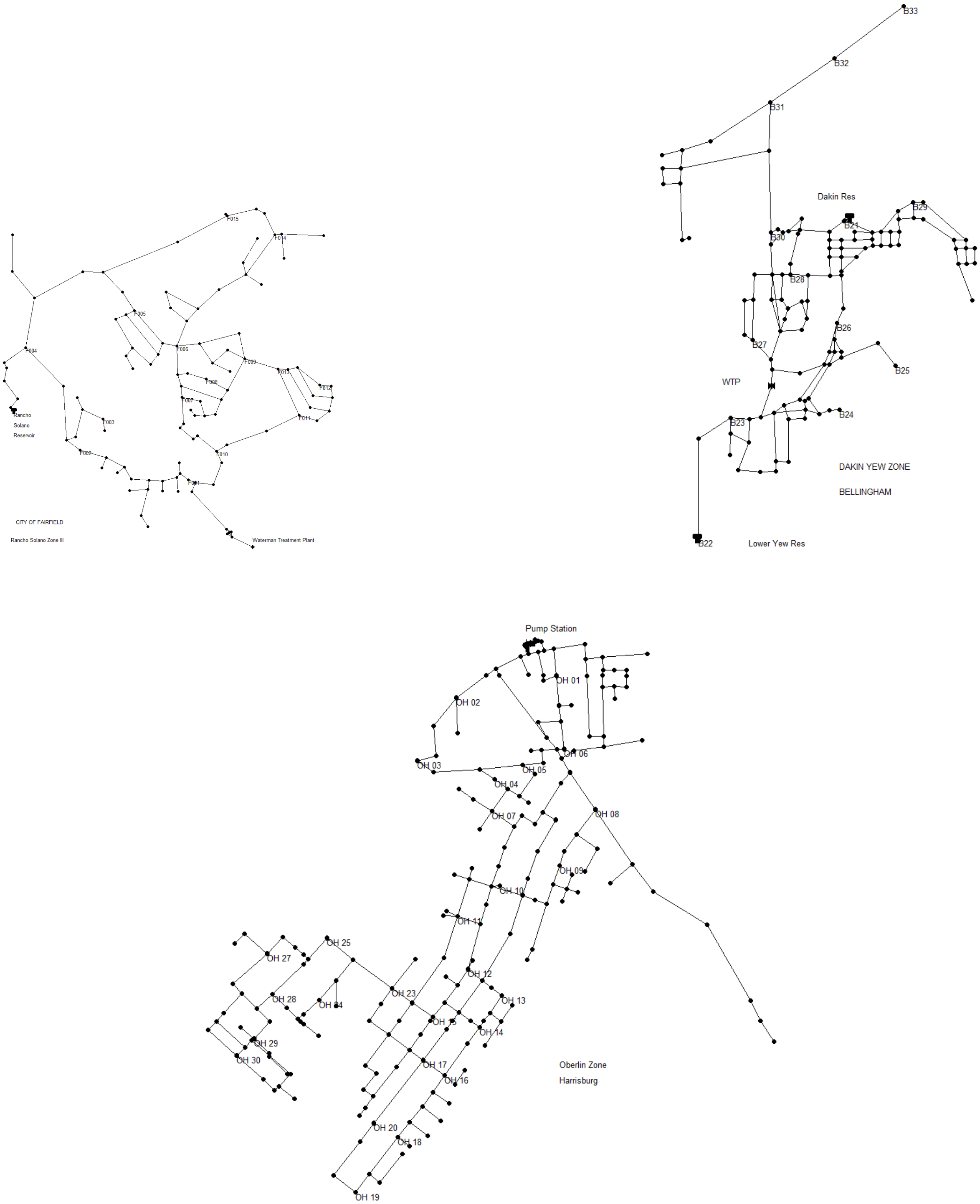

Figure 5.4: Network maps of the three water distribution systems: Fairfield (upper left), Bellingham (upper right), and Harrisburg (bottom).

`torch.linalg.lstsq` method[5] now supports automatic differentiation, allowing PyTorch to automatically backpropagate through the Algorithm 5.1 iterations, instead of using Theorem 5.4. We found that Algorithm 5.1 terminated with a small enough number of iterations that automatic differentiation was faster, so we opted to use this rather than the method from Theorem 5.4. We trained the IFN models to minimize the RMSE loss function by minimizing the RMSE loss function

$$\ell_{\text{rmse}} = \sqrt{\frac{1}{|\mathcal{E}_l|} \sum_{e \in \mathcal{E}} (f_e - \text{FN}_{h,\theta}(G,u)_e)^2}$$

5.8.4 Details on Baselines

We implemented all of the baselines by adapting Silva's code[6] from [111], refactoring some utility functions to decrease runtime. Following [111], we perform the following two data normalization steps:

(i) negative flows are converted into positive flows by flipping the orientation of the corresponding edges and replacing the entries of `f_true` with their absolute value, and

(ii) flows are proportionally normalized to the range $[0, 1]$ within each network.

After training with the normalized flows and computing the missing flow predictions, the predictions are denormalized before computing the testing RMSE.

Div The minimizing divergence baseline from [71] has a single hyperparameter, λ, from the regularization term $\lambda^2||f||_2^2$ in the loss function. We set $\lambda = 0.1$ for all networks and fractions of labeled edges by hand-tuning the parameter to the proper order of magnitude.

[5] https://pytorch.org/docs/stable/generated/torch.linalg.lstsq.html

[6] https://openreview.net/forum?id=l0V53bErniB

Bilevel Baselines All three of the bilevel baselines (Bil-MLP, Bil-GCN, and Bil-True) have several hyperparameters related to the bilevel optimization algorithm. For most of these parameters, we use the same settings as [111]: the number of iterations for the inner optimization problem is 300 during training and 3000 during evaluation, and the number of k-fold cross validation folds is 10; however, we increased the number of iterations of the outer optimization problem from 10 to 100, with an early stopping interval of 10, to ensure that the outer optimization problem was given sufficient time to converge. As with [111], we used a 2-layer MLP and GCN in Bil-MLP and Bil-GCN, respectively, but we increased the size of the hidden layer to 64.

Bil-True Like IFN, the baselines Bil-MLP and Bil-GCN train a model to predict edge weights from side information (if we interpret $\mathcal{Q}$ as a diagonal matrix of edge weights). We devised Bil-True as a third baseline to use the "ground-truth edge weights" instead of training a model. For water experiments, these ground-truth edge weights are given by (5.22). For the power experiments, we compute these edges weights from the AC active power flow equation: in a lossless AC power grid, active power flows f_{ij} on each edge $\{i, j\} \in \mathcal{E}$ are given by

$$f_{ij} = \frac{v_i v_j}{x_{ij} \tau_{ij}} \sin(\theta_i - \theta_j) \approx \frac{v_i v_j}{x_{ij} \tau_{ij}} (\theta_i - \theta_j) \tag{5.23}$$

where v_i, v_j are the voltage magnitudes on incident nodes, x_{ij} is the series reactance, τ_{ij} is the tap ratio, and θ_i, θ_j are the incident voltage angles. Since (5.23) is the constitutive relationship for AC power networks, examining its linear approximation in light of (5.15) suggests using $x_{ij}\tau_{ij}/v_i v_j$ as the regularizer weight on f_{ij}.

Bibliography

[1] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter. Differentiable convex optimization layers. In *Advances in Neural Information Processing Systems*, 2019. URL: https://arxiv.org/abs/1910.12430.

[2] M. Aigner. *A Course in Enumeration*. Springer, 2007, ISBN 9783540390350. doi:10.1007/978-3-540-39035-0.

[3] N. Ainsworth and S. Grijalva. A structure-preserving model and sufficient condition for frequency synchronization of lossless droop inverter-based AC networks. *IEEE Transactions on Power Systems*, 28(4):4310–4319, 2013. doi:10.1109/TPWRS.2013.2257887.

[4] V. V. Aleksandrov. On the accumulation of perturbations in the linear systems with two coordinates. *Vestnik MGU*, 3:67–76, 1968. (in Russian).

[5] M. Athans and P. L. Falb. *Optimal Control: An Introduction to the Theory and Its Applications*. McGraw-Hill, 1966, ISBN 0486453286.

[6] T. Athay, R. Podmore, and S. Virmani. A practical method for the direct analysis of transient stability. *IEEE Transactions on Power Apparatus and Systems*, 98(2):573–584, 1979. doi:10.1109/TPAS.1979.31940.

[7] S. Bai, J. Z. Kolter, and V. Koltun. Deep equilibrium models. In *Advances in Neural Information Processing Systems*, 2019. URL: https://arxiv.org/abs/1909.01377.

[8] P. Battaglia, R. Pascanu, M. Lai, D. Jimenez Rezende, and K. Kavukcuoglu. Interaction networks for learning about objects, relations and physics. In *Advances in Neural Information Processing Systems*. 2016.

[9] G. Berkolaiko, N. Duffield, M. Ettehad, and K. Manousakis. Graph reconstruction from path correlation data. *Inverse Problems*, 35(1):015001, 2018. doi:10.1088/1361-6420/aae798.

[10] J. T. Betts. *Pratical Methods for Optimal Control and Estimation Using Nonlinear Programming*. SIAM, 2010, ISBN 978-0-89871-688-7.

[11] A. B. Birchfield, T. Xu, K. M. Gegner, K. S. Shetye, and T. J. Overbye. Grid structural characteristics as validation criteria for synthetic networks. *IEEE Transactions on power systems*, 32(4):3258–3265, 2016. doi:10.1109/TPWRS.2016.2616385.

[12] F. Blanchini and S. Miani. *Set-Theoretic Methods in Control*. Springer, 2015, ISBN 9783319179322.

[13] P. Bonami, A. Lodi, A. Tramontani, and S. Wiese. On mathematical programming with indicator constraints. *Mathematical Programming*, 151:191–223, 2015. doi:10.1007/s10107-015-0891-4.

[14] L. Böttcher, N. Antulov-Fantulin, and T. Asikis. AI Pontryagin or how artificial neural networks learn to control dynamical systems. *Nature Communications*, 13(1):1–9, 2022. doi:10.1038/s41467-021-27590-0.

[15] L. Böttcher, H. Wolf, B. Jung, P. Lutat, M. Trageser, O. Pohl, A. Ulbig, and M. Grohe. Solving AC power flow with graph neural networks under realistic constraints. *arXiv preprint arXiv:2204.07000*, 2022. URL: https://arxiv.org/abs/2204.07000, doi:10.48550/ARXIV.2204.07000.

[16] S. Boyd, S.-J. Kim, L. Vandenberghe, and A. Hassibi. A tutorial on geometric programming. *Optimization and Engineering*, 8(1):67–127, 2007. doi:10.1007/s11081-007-9001-7.

[17] A. Bressan and B. Piccoli. *Introduction to the Mathematical Theory of Control*. American Institute of Mathematical Sciences, 2007, ISBN 1-60133-002-2.

[18] J. C. Bronski, T. Carty, and L. DeVille. Configurational stability for the Kuramoto–Sakaguchi model. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(10):103109, 2018. doi:10.1063/1.5029397.

[19] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

[20] F. Bullo. *Lectures on Network Systems*. Kindle Direct Publishing, 1.4 edition, July 2020, ISBN 978-1986425643. With contributions by J. Cortés, F. Dörfler, and S. Martínez. URL: http://motion.me.ucsb.edu/book-lns.

[21] F. Bullo. *Contraction Theory for Dynamical Systems*. Kindle Direct Publishing, 1.0 edition, 2022, ISBN 979-8836646806. URL: http://motion.me.ucsb.edu/book-ctds.

[22] F. Bullo. *Lectures on Network Systems*. Kindle Direct Publishing, 1.6 edition, January 2022, ISBN 978-1986425643. URL: http://motion.me.ucsb.edu/book-lns.

[23] M. C. Chandorkar, D. M. Divan, and R. Adapa. Control of parallel connected inverters in standalone AC supply systems. *IEEE Transactions on Industry Applications*, 29(1):136–143, 1993. doi:10.1109/28.195899.

[24] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, 2018. URL: https://arxiv.org/abs/1806.07366.

[25] F. L. Chernousko and A. A. Lyubushin. Method of successive approximations for solution of optimal control problems. *Optimal Control Applications and Methods*, 3(2):101–114, 1982. doi:10.1002/oca.4660030201.

[26] H.-D. Chiang, C. C. Chu, and G. Cauley. Direct stability analysis of electric power systems using energy functions: Theory, applications, and perspective. *Proceedings of the IEEE*, 83(11):1497–1529, 1995. doi:10.1109/5.481632.

[27] M. Coates, R. Castro, R. Nowak, M. Gadhiok, R. King, and Y. Tsang. Maximum likelihood network topology identification from edge-based unicast measurements. In *ACM SIGMETRICS Performance Evaluation Review*, pages 11–20, 2002. doi:10.1145/511399.511337.

[28] M. Coates, A. O. Hero III, R. Nowak, and B. Yu. Internet tomography. *IEEE Signal Processing Magazine*, 19(3):47–65, 2002. doi:10.1109/79.998081.

[29] T. Coletta, R. Delabays, I. Adagideli, and P. Jacquod. Topologically protected loop flows in high voltage AC power grids. *New Journal of Physics*, 18(10):103042, 2016. doi:10.1088/1367-2630/18/10/103042.

[30] G. Como. On resilient control of dynamical flow networks. *Annual Reviews in Control*, 43:80–90, 2017. doi:10.1016/j.arcontrol.2017.01.001.

[31] B. A. Conway. A survey of methods available for the numerical optimization of continuous dynamic systems. *Journal of Optimization Theory and Applications*, 152(2):271–306, 2012. doi:10.1007/s10957-011-9918-z.

[32] P. E. Crouch, F. Lamnabhi-Lagarrigue, and A. J. van der Schaft. Adjoint and Hamiltonian input-output differential equations. *IEEE Transactions on Automatic Control*, 40(4):603–615, 1995. doi:10.1109/9.376115.

[33] A. C. Davison and D. V. Hinkley. *Boostrap Methods and Their Application*. Cambridge University Press, 1999, ISBN 9781107266827.

[34] G. De Pasquale, K. D. Smith, F. Bullo, and M. E. Valcher. Dual seminorms, ergodic coefficients, and semicontraction theory. *Technical Report*, 2022. doi:10.48550/arXiv.2201.03103.

[35] F. De Smet and D. Aeyels. Partial entrainment in the finite Kuramoto–Sakaguchi model. *Physica D: Nonlinear Phenomena*, 234(2):81–89, 2007. doi:10.1016/j.physd.2007.06.025.

[36] M. H. DeGroot and M. J. Schervish. *Probability and Statistics*. Pearson Education, 4th edition, 2012, ISBN 9780321500465.

[37] P. Van den Driessche and J. Watmough. Reproduction numbers and sub-threshold endemic equilibria for compartmental models of disease transmission. *Mathematical Biosciences*, 180(1):29–48, 2002. doi:10.1016/S0025-5564(02)00108-6.

[38] O. Diekmann, J. A. P. Heesterbeek, and J. A. J. Metz. On the definition and the computation of the basic reproduction ratio R_0 in models for infectious diseases in heterogeneous populations. *Journal of Mathematical Biology*, 28(4):365–382, 1990. doi:10.1007/BF00178324.

[39] G. M. U. Din and A. K. Marnerides. Short term power load forecasting using deep neural networks. In *2017 International conference on computing, networking and communications (ICNC)*, pages 594–598. IEEE, 2017. doi:10.1109/ICCNC.2017.7876196.

[40] F. Dörfler and F. Bullo. Synchronization and transient stability in power networks and non-uniform Kuramoto oscillators. *SIAM Journal on Control and Optimization*, 50(3):1616–1642, 2012. doi:10.1137/110851584.

[41] P. G. Doyle and J. L. Snell. *Random Walks and Electric Networks*. Mathematical Association of America, 1984, ISBN 0883850249. URL: https://math.dartmouth.edu/~doyle/docs/walks/walks.pdf.

[42] N. G. Duffield, J. Horowitz, F. Lo Presti, and D. Towsley. Multicast topology inference from measured end-to-end loss. *IEEE Transactions on Information Theory*, 48(1):26–45, 2002. doi:10.1109/18.971737.

[43] W. E. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 1(5):1–11, 2017. doi:10.1007/s40304-017-0103-z.

[44] L. El Ghaoui, F. Gu, B. Travacca, A. Askari, and A. Tsai. Implicit deep learning. *SIAM Journal on Mathematics of Data Science*, 3(3):930–958, 2021. doi:10.1137/20M1358517.

[45] E. John Finnemore and Joseph B. Franzini. *Fluid mechanics with engineering applications*. McGraw-Hill, tenth edition, 2002, ISBN 007112196X.

[46] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. In *Canadian Journal of Mathematics*, volume 8, pages 399–404, 1956. doi:10.4153/CJM-1956-045-5.

[47] S. W. Fung, H. Heaton, Q. Li, D. McKenzie, S. Osher, and W. Yin. Fixed point networks: Implicit depth models with Jacobian-free backprop, 2021. ArXiv e-print. URL: https://arxiv.org/abs/2103.12803.

[48] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pages 1263–1272. PMLR, 2017.

[49] A. Gkelias, L. Ma, K. K. Leung, A. Swami, and D. Towsley. Robust and efficient monitor placement for network tomography in dynamic networks. *IEEE/ACM Transactions on Networking*, 25(3):1732–1745, 2017. doi:10.1109/TNET.2016.2642185.

[50] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1, March 2014. URL: http://cvxr.com/cvx.

[51] M. Green and D. J. N. Limebeer. *Linear Robust Control*. Prentice Hall, 1995, ISBN 0131022784.

[52] C. Grigg et al. The IEEE Reliability Test System-1996. A report prepared by the Reliability Test System Task Force of the Application of Probability Methods Subcommittee. *IEEE Transactions on Power Systems*, 14(3):1010–1020, 1999. doi:10.1109/59.780914.

[53] L. L. Grigsby, editor. *Power System Stability and Control*. CRC Press, 3 edition, 2012, ISBN 9781439883204.

[54] D. Groß, J. S. Brouillon, and F. Dörfler. The effect of transmission-line dynamics on grid-forming dispatchable virtual oscillator control. *IEEE Transactions on Control of Network Systems*, 6(3):1148–1160, 2019. doi:10.1109/TCNS.2019.2921347.

[55] NERC Steering Group. Technical Analysis of the August 14, 2003, Blackout: What Happened, Why, and What Did We Learn? Technical report, North American Electric Reliability Council, Princeton Forrestal Village, Princeton, NJ, USA, July 2004.

[56] F. Gu, H. Chang, W. Zhu, S. Sojoudi, and L. El Ghaoui. Implicit graph neural networks. In *Advances in Neural Information Processing Systems*, 2020. URL: https://arxiv.org/abs/2009.06211.

[57] J. M. Guerrero, J. C. Vasquez, J. Matas, L. G. de Vicuna, and M. Castilla. Hierarchical control of droop-controlled AC and DC microgrids–a general approach toward standardization. *IEEE Transactions on Industrial Electronics*, 58(1):158–172, 2011. doi:10.1109/TIE.2010.2066534.

[58] M. H. Gunes and K. Sarac. Analyzing router responsiveness to active measurement probes. In *International Conference on Passive and Active Network Measurement*, pages 23–32, 2009. doi:10.1007/978-3-642-00975-4_3.

[59] W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[60] M. Hayhoe, F. Barreras, and V. M. Preciado. Multitask learning and nonlinear optimal control of the COVID-19 outbreak: A geometric programming approach. *Annual Reviews in Control*, 2021. doi:10.1016/j.arcontrol.2021.04.014.

[61] H. Heaton, D. McKenzie, Q. Li, S. W. Fung, S. Osher, and W. Yin. Learn to predict equilibria via fixed point networks. *arXiv preprint arXiv:2106.00906*, 2021. doi:10.48550/arXiv.2106.00906.

[62] E. Hernadez, S. Hoagland, and L. Ormsbee. Water distribution database for research applications. In *World Environmental and Water Resources Congress*, pages 465–474, 2016. doi:10.1061/9780784479865.049.

[63] B. Holbert, S. Tati, S. Silvestri, T. La Porta, and A. Swami. Network topology inference with partial information. *IEEE Transactions on Network and Service Management*, 12(3):406–419, 2015. doi:10.1109/TNSM.2015.2451032.

[64] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 2nd edition, 2012, ISBN 0521548233.

[65] A. R. Hota, J. Godbole, and P. E. Paré. A closed-loop framework for inference, prediction, and control of SIR epidemics on networks. *IEEE Transactions on Network Science and Engineering*, 8(3):2262–2278, 2021. doi:10.1109/TNSE.2021.3085866.

[66] J. A. Jacquez and C. P. Simon. Qualitative theory of compartmental systems. *SIAM Review*, 35(1):43–79, 1993. doi:10.1137/1035003.

[67] S. Jafarpour, A. Davydov, A. V. Proskurnikov, and F. Bullo. Robust implicit networks via non-Euclidean contractions. In *Advances in Neural Information Processing Systems*, December 2021. doi:10.48550/arXiv.2106.03194.

[68] S. Jafarpour, E. Y. Huang, K. D. Smith, and F. Bullo. Flow and elastic networks on the n-torus: Geometry, analysis and computation. *SIAM Review*, 2019. Submitted. URL: https://arxiv.org/pdf/1901.11189v3.pdf.

[69] S. Jafarpour, E. Y. Huang, K. D. Smith, and F. Bullo. Flow and elastic networks on the n-torus: Geometry, analysis and computation. *SIAM Review*, 64(1):59–104, 2022. doi:10.1137/18M1242056.

[70] N. Janssens and A. Kamagate. Loop flows in a ring AC power system. *International Journal of Electrical Power & Energy Systems*, 25(8):591–597, 2003. doi:10.1016/S0142-0615(03)00017-6.

[71] J. Jia, M. T. Schaub, S. Segarra, and A. R. Benson. Graph-based semi-supervised & active learning for edge flows. In *ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, page 761–771, July 2019. doi:10.1145/3292500.3330872.

[72] X. Jin, W.-P. K. Yiu, S.-H. G. Chan, and Y. Wang. Network topology inference based on end-to-end measurements. *IEEE Journal on Selected Areas in Communications*, 24(12):2182–2195, 2006. doi:10.1109/JSAC.2006.884016.

[73] J. Jo, J. Baek, S. Lee, D. Kim, M. Kang, and S. J. Hwang. Edge representation learning with hypergraphs. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 7534–7546. Curran Associates, Inc., 2021. URL: https://proceedings.neurips.cc/paper/2021/file/3def184ad8f4755ff269862ea77393dd-Paper.pdf.

[74] H. B. Keller. *Numerical Methods for Two-Point Boundary-Value Problems.* Dover Publications, 2018, ISBN 9780486828343.

[75] H. J. Kelley, R. E. Kopp, and H. G. Moyer. Successive approximation techniques for trajectory optimization. Technical report, Grumman Aircraft Engineering Corp, Bethpage NY, 1961. URL: https://apps.dtic.mil/sti/citations/AD0268321.

[76] H. K. Khalil. *Nonlinear Systems.* Prentice Hall, 3 edition, 2002, ISBN 0130673897.

[77] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[78] K. A. Klise, R. Murray, and T. Haxton. An overview of the water network tool for resilience (WNTR). In *Proceedings of the 1st International WDSA/CCWI Joint Conference*, volume 1, 2018.

[79] O. Kouba and D. S. Bernstein. What is the adjoint of a linear system? [lecture notes]. *IEEE Control Systems*, 40(3):62–70, 2020. doi:10.1109/MCS.2020.2976389.

[80] I. A. Krylov and F. L. Chernous'ko. On a method of successive approximations for the solution of problems of optimal control. *USSR Computational Mathematics and Mathematical Physics*, 2(6):1371–1382, 1963. doi:10.1016/0041-5553(63)90353-7.

[81] S. Kundu, W. Du, S. P. Nandanoori, F. Tuffner, and K. Schneider. Identifying parameter space for robust stability in nonlinear networks: A microgrid application. In *American Control Conference*, pages 3111–3116, July 2019. doi:10.23919/ACC.2019.8814324.

[82] S. Kundu, S. P. Nandanoori, K. Kalsi, S. Geng, and I. A. Hiskens. Distributed barrier certificates for safe operation of inverter-based microgrids. In *American Control Conference*, pages 1042–1047, July 2019. doi:10.23919/ACC.2019.8815296.

[83] P. Kundur. *Power System Stability and Control*. McGraw-Hill, 1994, ISBN 007035958X.

[84] S. Lee, G. Chowell, and C. Castillo-Chávez. Optimal control for pandemic influenza: the role of limited antiviral treatment and isolation. *Journal of Theoretical Biology*, 265:136–150, 2010. doi:10.1016/j.jtbi.2010.04.003.

[85] S. Lenhart and J. T. Workman. *Optimal Control Applied to Biological Models*. Chapman and Hall, 2007, ISBN 9780429138058.

[86] Q. Li, L. Chen, C. Tai, and W. E. Maximum principle based algorithms for deep learning. *Journal of Machine Learning Research*, 18(165):1–29, 2018. URL: http://jmlr.org/papers/v18/17-653.html.

[87] Q. Li and S. Hao. An optimal control approach to deep learning and applications to discrete-weight neural networks. In *International Conference on Machine Learning*, pages 2985–2994, 2018. URL: https://proceedings.mlr.press/v80/li18b.html.

[88] Y. Li, R. Yu, C. Shahabi, and Y. Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018. URL: https://openreview.net/forum?id=SJiHXGWAZ.

[89] M. Lippi, M. Bertini, and P. Frasconi. Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. *IEEE Transactions on Intelligent Transportation Systems*, 14(2):871–882, 2013. doi:10.1109/TITS.2013.2247040.

[90] M. Luckie, Y. Hyun, and B. Huffaker. Traceroute probe method and forward ip path inference. In *ACM SIGCOMM Conference on Internet Measurement*, pages 311–324, 2008. doi:10.1145/1452520.1452557.

[91] L. Ma, T. He, A. Swami, D. Towsley, and K. K. Leung. Network capability in localizing node failures via end-to-end path measurements. *IEEE/ACM Transactions on Networking*, 25(1):434–450, 2016. doi:10.1109/TNET.2016.2584544.

[92] V. S. Mai, A. Battou, and K. Mills. Distributed algorithm for suppressing epidemic spread in networks. *IEEE Control Systems Letters*, 2(3):555–560, 2018. doi:10.1109/LCSYS.2018.2844118.

[93] M. McAsey, L. Mou, and W. Han. Convergence of the forward-backward sweep method in optimal control. *Computational Optimization and Applications*, 53(1):207–226, 2012. doi:10.1007/s10589-011-9454-7.

[94] P. McCullagh and J. Kolassa. Cumulants. *Scholarpedia*, 4(3):4699, 2009. doi:10.4249/scholarpedia.4699.

[95] S. K. Mitter. Successive approximation methods for the solution of optimal control problems. *Automatica*, 3(3-4):135–149, 1966. doi:10.1016/0005-1098(66)90009-4.

[96] E. Di Nardo and G. Guarino. kstatistics: Unbiased estimators for cumulant products, 2019. R package version 1.0. URL: https://CRAN.R-project.org/package=kStatistics.

[97] E. Di Nardo, G. Guarino, and D. Senato. A new method for fast computing unbiased estimators of cumulants. *Statistics and Computing*, 19(2):155, 2009. doi:10.1007/s11222-008-9080-0.

[98] J. Ni, H. Xie, S. Tatikonda, and Y. R. Yang. Efficient and dynamic routing topology inference from end-to-end measurements. *IEEE/ACM Transactions on Networking*, 18(1):123–135, 2009. doi:10.1109/TNET.2009.2022538.

[99] C. Nowzari, V. M. Preciado, and G. J. Pappas. Optimal resource allocation for control of networked epidemic models. *IEEE Transactions on Control of Network Systems*, 4:159–169, 2017. doi:10.1109/TCNS.2015.2482221.

[100] M. Ogura, M. Kishida, and J. Lam. Geometric programming for optimal positive linear systems. *IEEE Transactions on Automatic Control*, 65(11):4648–4663, 2020. doi:10.1109/TAC.2019.2960697.

[101] V. M. Preciado, M. Zargham, C. Enyioha, A. Jadbabaie, and G. J. Pappas. Optimal resource allocation for network protection against spreading processes. *IEEE Transactions on Control of Network Systems*, 1(1):99–108, 2014. doi:10.1109/TCNS.2014.2310911.

[102] M. G. Rabbat, M. J. Coates, and R. D. Nowak. Multiple-source Internet tomography. *IEEE Journal on Selected Areas in Communications*, 24(12):2221–2234, 2006. doi:10.1109/JSAC.2006.884020.

[103] A. V. Rao. A survey of numerical methods for optimal control. *Advances in the Astronautical Sciences*, 135(1):497–528, 2009.

[104] S. Ratnasamy and S. McCanne. Inference of multicast routing trees and bottleneck bandwidths using end-to-end measurements. In *IEEE Conf. on Computer Communications*, pages 353–360, 1999. doi:10.1109/INFCOM.1999.749302.

[105] M. Revay, R. Wang, and I. R. Manchester. Lipschitz bounded equilibrium networks. 2020. URL: https://arxiv.org/abs/2010.01732.

[106] S. M. Rinaldi, J. P. Peerenboom, and T. K. Kelly. Identifying, understanding, and analyzing critical infrastructure interdependencies. *IEEE Control Systems Magazine*, 21(6):11–25, 2001. doi:10.1109/37.969131.

[107] T. Mitchell Roddenberry, Nicholas Glaze, and Santiago Segarra. Principled simplicial neural networks for trajectory prediction. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9020–9029. PMLR, 18–24 Jul 2021. URL: https://proceedings.mlr.press/v139/roddenberry21a.html.

[108] R. E. Rowthorn, R. Laxminarayan, and C. A. Gilligan. Optimal control of epidemics in metapopulations. *Journal of the Royal Society Interface*, 6(41):1135–1144, 2009. doi:10.1098/rsif.2008.0402.

[109] A. Sabnis, R. K. Sitaraman, and D. Towsley. OCCAM: An optimization based approach to network inference. *ACM SIGMETRICS Performance Evaluation Review*, 46(2):36–38, 2019. doi:10.1145/3305218.3305232.

[110] H. Sakaguchi and Y. Kuramoto. A soluble active rotater model showing phase transitions via mutual entertainment. *Progress of Theoretical Physics*, 76(3):576–581, 1986. doi:10.1143/PTP.76.576.

[111] A. Silva, F. Kocayusufoglu, S. Jafarpour, F. Bullo, A. Swami, and A. K. Singh. Combining physics and machine learning for network flow estimation. In *International Conference on Learning Representations*, Online, May 2021. URL: https://openreview.net/forum?id=l0V53bErniB.

[112] J. W. Simpson-Porco, F. Dörfler, and F. Bullo. Synchronization and power sharing for droop-controlled inverters in islanded microgrids. *Automatica*, 49(9):2603–2611, 2013. doi:10.1016/j.automatica.2013.05.018.

[113] A. K. Singh, Ibraheem, S. Khatoon, M. Muazzam, and D. K. Chaturvedi. Load forecasting techniques and methodologies: A review. In *2012 2nd International Conference on Power, Control and Embedded Systems*. IEEE, 2012. doi:10.1109/ICPCES.2012.6508132.

[114] K. D. Smith. PyMoments: A Python toolkit for unbiased estimation of multivariate statistical moments, 2020. URL: https://github.com/KevinDalySmith/PyMoments.

[115] K. D. Smith. A tutorial on multivariate k-statistics and their computation, 2020. URL: http://arxiv.org/pdf/2005.08373.

[116] K. D. Smith and F. Bullo. Contractivity of the method of successive approximations for optimal control. *IEEE Control Systems Letters*, November 2022. To appear. doi:10.48550/arXiv.2209.10046.

[117] K. D. Smith and F. Bullo. Convex optimization of the basic reproduction number. *IEEE Transactions on Automatic Control*, 2022. To appear. URL: https://arxiv.org/abs/2109.07643.

[118] K. D. Smith, S. C. Hsiung, C. White, C. G. Lowe, and C. M. Clark. Stochastic modeling and control for tracking the periodic movement of marine animals via AUVs. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3101–3107. IEEE, 2016. doi:10.1109/IROS.2016.7759480.

[119] K. D. Smith, S. Jafarpour, and F. Bullo. Transient stability of droop-controlled inverter networks with operating constraints. *IEEE Transactions on Automatic Control*, 67(2):633–645, 2022. doi:10.1109/TAC.2021.3053552.

[120] K. D. Smith, S. Jafarpour, A. Swami, and F. Bullo. Topology inference with multivariate cumulants: The Möbius inference algorithm. *IEEE/ACM Transactions on Networking*, 30(5):2102–2116, 2022. doi:10.1109/TNET.2022.3164336.

[121] K. D. Smith, F. Seccamonte, A. Swami, and F. Bullo. Physics-informed implicit representations of equilibrium network flows. In *Advances in Neural Information Processing Systems*, 2022. To appear.

[122] K. D. Smith and K. Studarus. Limited-knowledge economic dispatch prediction using bayesian averaging of single-node models. In *2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*. IEEE, 2018. doi:10.1109/PMAPS.2018.8440564.

[123] V. L. J. Somers and I. R. Manchester. Sparse resource allocation for control of spreading processes via convex optimization. *IEEE Control Systems Letters*, 5(2):547–552, 2020. doi:10.1109/LCSYS.2020.3003401.

[124] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with Rocketfuel. *ACM SIGCOMM Computer Communication Review*, 32(4):133–145, 2002. doi:10.1145/964725.633039.

[125] B. Staude, S. Rotter, and S. Grün. CuBIC: cumulant based inference of higher-order correlations in massively parallel spike trains. *Journal of Computational Neuroscience*, 29(1-2):327–350, 2010. doi:10.1007/s10827-009-0195-x.

[126] O. V. Stryk. Numerical solution of optimal control problems by direct collocation. In *Optimal Control*, pages 129–143. Springer, 1993. doi:10.1007/978-3-0348-7539-4_10.

[127] P. Tabuada and B. Gharesifard. Universal approximation power of deep residual neural networks through the lens of control. *IEEE Transactions on Automatic Control*, 2022. doi:10.1109/TAC.2022.3190051.

[128] J. A. Torres, S. Roy, and Y. Wan. Sparse resource allocation for linear network spread dynamics. *IEEE Transactions on Automatic Control*, 62(4):1714–1728, 2017. doi:10.1109/TAC.2016.2593895.

[129] P. Varaiya, F. F. Wu, and R.-L. Chen. Direct methods for transient stability analysis of power systems: Recent results. *Proceedings of the IEEE*, 73(12):1703–1715, 1985. doi:10.1109/PROC.1985.13366.

[130] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. URL: https://arxiv.org/abs/1710.10903.

[131] N. K. Vishnoi. $Lx = b$, Laplacian solvers and their algorithmic applications. *Theoretical Computer Science*, 8(1-2):1–141, 2013. doi:10.1561/0400000054.

[132] T. L. Vu and K. Turitsyn. Lyapunov functions family approach to transient stability assessment. *IEEE Transactions on Power Systems*, 31(2):1269–1277, 2016. doi:10.1109/TPWRS.2015.2425885.

[133] T. L. Vu and K. Turitsyn. A framework for robust assessment of power grid stability and resiliency. *IEEE Transactions on Automatic Control*, 62(3):1165–1177, 2017. doi:10.1109/TAC.2016.2579743.

[134] E. Winston and J. Z. Kolter. Monotone operator equilibrium networks. In *Advances in Neural Information Processing Systems*, 2020. URL: https://arxiv.org/abs/2006.08591.

[135] K. Xi, H. X. Lin, C. Shen, and J. H. Van Schuppen. Multilevel power-imbalance allocation control for secondary frequency control of power systems. *IEEE Transactions on Automatic Control*, 65(7):2913–2928, 2020. doi:10.1109/TAC.2019.2934014.

[136] Maosheng Yang, Elvin Isufi, Michael T. Schaub, and Geert Leus. Simplicial convolutional filters. *arXiv prepint arXiv:2201.11720*, 2022. URL: https://arxiv.org/abs/2201.11720, doi:10.48550/ARXIV.2201.11720.

[137] B. Yao, R. Viswanathan, F. Chang, and D. Waddington. Topology inference in the presence of anonymous routers. In *IEEE Conf. on Computer Communications*, pages 353–363, 2003. doi:10.1109/INFCOM.2003.1208687.

[138] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 5668–5675, 2019. doi:10.1609/aaai.v33i01.33015668.

[139] M. K. S. Yeung and S. H. Strogatz. Time delay in the Kuramoto model of coupled oscillators. *Physical Review Letters*, 82(3):648, 1999. doi:10.1103/PhysRevLett.82.648.

[140] D. Zhang, T. Zhang, Y. Lu, Z. Zhu, and B. Dong. You only propagate once: Accelerating adversarial training via maximal principle. *Advances in Neural Information Processing Systems*, 32, 2019. doi:10.48550/arXiv.1905.00877.

[141] X. Zhang and C. Phillips. A survey on selective routing topology inference through active probing. *IEEE Communications Surveys & Tutorials*, 14(4):1129–1141, 2011. doi:10.1109/SURV.2011.081611.00040.

[142] Y. Zhang, D. Hatzinakos, and A. N. Venetsanopoulos. Bootstrapping techniques in the estimation of higher-order cumulants from short data records. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, volume 4, pages 200–203, 1993. doi:10.1109/ICASSP.1993.319629.

[143] L. Zhu and D. Hill. Stability analysis of power systems: A network synchronization perspective. *SIAM Journal on Control and Optimization*, 56(3):1640–1664, 2018. doi:10.1137/17M1118646.

[144] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas. MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on Power Systems*, 26(1):12–19, 2011. doi:10.1109/TPWRS.2010.2051168.