Structured Control and Learning for Sustainable Power Systems

Wenqi Cui

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

Doctor of Philosophy

University of Washington

2024

Reading Committee:
Baosen Zhang, Chair
Daniel S. Kirschen
Maryam Fazel

Program Authorized to Offer Degree:
Electrical and Computer Engineering

University of Washington

Abstract

Structured Control and Learning for Sustainable Power Systems

Wenqi Cui

Chair of the Supervisory Committee:
Baosen Zhang
Electrical and Computer Engineering

With decarbonization efforts in renewable integration and electrification, the electric grid needs to adapt and serve a larger system that is becoming more distributed, having less inertia, and facing more uncertainties. These changes have reduced the safety margins of the grid and significantly increased the costs of risk management. Machine learning tools can potentially unlock design freedoms found in the increased controllability from inverter-interfaced resources (e.g., solar, wind, and electric vehicles), and reshape the landscape of power systems for more efficient operations. However, such algorithms typically do not provide guarantees about safety-critical constraints, making them difficult to implement in practice.

The dissertation proposes to bridge the gap between learning and safety-critical constraints through structured neural networks guided by control theory and the physics of power systems. Using Lyapunov theory, we extract stabilizing controller structures for transient stability problems, and parameterize the structures by neural networks. On this basis, we design Neural-PI controllers to further achieve provable guarantees on optimal resource allocation and frequency restoration at the steady state. In addition, we propose a modular approach for transient stability analysis with lossy transmission lines. This provides a simple yet effective approach to optimize control efforts with guaranteed stability regions.

The structured approach for learning-based control provides end-to-end guarantees that are independent of the learning process, which in turn provides large flexibility for learning algorithm

design. To relieve the burden of centralized coordination in voltage control, we propose a decentralized safe learning approach to train local neural network controllers at each node in a model-free setting. To overcome key barriers on the requirement of a large number of system interactions to learn a good control policy, we develop a sample-efficient trajectory generation algorithm that adapts to the distributional shift of trajectories resulting from updated control policies and also extends to partially observed systems.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

My PhD journey has been wonderful and enjoyable, thanks to the collaboration, guidance, and encouragement of many remarkable and generous individuals who have shaped both me and my research. First and foremost, I would like to express my deepest appreciation to my awesome advisor, Professor Baosen Zhang, for his invaluable guidance and support throughout my PhD studies. I am incredibly fortunate to learn from Baosen through every aspect of research, including and not limited to the insights of selecting the right and interesting research questions, building theoretical foundations, presenting results, and fostering collaboration. It was always enjoyable and inspiring to discuss with Baosen. His innovative perspectives and the philosophy of looking at the simplest settings with the most intuitions, profoundly shaped the way I approach academic problems. In addition, I'm extremely grateful to Baosen for his endless support and encouragement for me to try internships and explore different research directions. These all helped me find my enthusiasm and fueled my passion for an academic career. I am profoundly thankful, and Baosen will always be an outstanding role model for my future endeavors.

I would also like to thank Professor Daniel Kirschen, who served on the committee for all milestones of my PhD study and also acted as an advisor in many ways. Through group meetings and lecturing, I learned immensely from his insightful suggestions and expertise in power systems. His passion for teaching and education, as well as his vision for addressing real-world challenges in power systems, will remain lifelong inspirations to me.

I extend my sincere thanks to my committee member, Professor Maryam Fazel, for providing insightful feedback on my research and for her extraordinary lectures on convex optimization. I would like to thank Professor Mehran Mesbahi for serving as the Graduate School Representative and leading the vibrant community Control + X community. Interacting with the Control + X

community has been a significant and enriching part of my PhD journey, and I am deeply grateful to Professor Mesbahi for his invaluable help and feedback in research, presentations, and career development. Additionally, I want to thank Professors Sam Burden and Brian Johnson for serving on my qualifying exam committee. I would also like to thank Professor Amir Taghvaei for his feedback on research and for the opportunity to deliver a guest lecture for the course EE583 Nonlinear Systems and Control.

It has been a wonderful pleasure to work with an amazing group of labmates and collaborators at UW. I would like to express my gratitude to Yuanyuan Shi, who introduced me to reinforcement learning at the beginning of my PhD journey and with whom I maintained consistent collaborations. She has been a great friend and mentor, and I deeply appreciate her tremendous suggestions in research, life, and career development. I would like to thank Yan Jiang, who taught me a lot about control theory and how to present results in a mathematically rigorous way. I want to thank Jiayi Li for exploring lots of new directions together. Additionally, I am grateful to Yize Chen, Daniel Tabas, Ling Zhang, Zixiao Ma, Trager Joswig-Jones, Matt Motoki, Atinuke Ademola-Idowu, Chase Dowling, and Hao Wang for all the stimulating conversations and discussions. I would like to extend my sincere thanks to Lane Smith, who taught me a lot about how to teach a class and interact with students. It was also a great pleasure taking many courses together with Lane, Gord Stephen, and Mareldi Ahumada-Parás. Thank you for all the enjoyable interactions, even during online courses after COVID. I also want to express my appreciation for UW Clean Energy Institute, which provides generous support and diverse opportunities to interact with the broader power systems community.

I was also fortunate to experience fantastic internships at Microsoft Research and meet wonderful collaborators outside of UW. I would like to thank my Mentor, Weiwei Yang, who taught me a lot from the perspectives of machine learning, data science, and neural computing. I deeply appreciate the freedom she provided for me to explore different topics during my internships, and the diverse resources she has provided both inside and outside the Living Interface group. I am

grateful to Kate Lytvynets for her hands-on support every day and for brightening my internship with many interesting events. I am also grateful for all of the conversations and collaborations with the other interns and researchers at Microsoft. In addition, I would like to thank Professor Jorge Cortés for providing numerous inspirational discussions and suggestions. I would like to extend my sincere thanks to Linbin Huang, who taught me data-driven control and brought many interesting discussions at the intersection of control, learning, and inverter-based systems. I am also thankful to Guanya Shi, who taught me adaptive control and lots of interesting applications in robotics.

Thanks to everyone else who made my graduate school incredibly enjoyable. These wonderful memories will stay with me for life. This includes Jingyuan Li, Feifei Yang, Yangwei Shi, Kun Su, Yang Zheng, Jinglin Xiang, Xiulong Liu, Yujia Liu, Mingfei Chen, Hao Yin, Mengyuan Wang, Sitong Zhou, Jing Wang, Tianyan Zhou, Yue Sun, Cong Chen, Jiaqi Chen, Zhe Yang, Yuqi Zhou, Xin Chen, Bingqing Chen, Meiyi Li, Nan Shang, Lixin Zhang, Zechen Zhang, Lu Mi. You have been a source of joy and encouragement in many moments, and I have learned a lot of valuable experiences from you. I want to thank all the other alumni in the group of power and power systems, including Qinghu Tang, Bosong Li, Yao Long, Ryan Elliott, Qingchun Hou, Hao Li, Nina Vincent, Jackie Baum, Trisha Ray, Weiqian Cai, Soham Dutta, Rahul Mallik, Bolun Xu, Yury Dvorkin. You brightened my lab time and were always happy to provide suggestions throughout my PhD journey. Thank you all from the bottom of my heart.

Lastly, I am profoundly grateful to my parents. Your unwavering support, encouragement, and belief in me have shaped me into who I am today. Your influence has been instrumental in every step of my academic journey, and I cannot thank you enough for your endless love and faith in me.

DEDICATION

To everybody loves me

Chapter 1

INTRODUCTION

1.1 Motivation

Increasing the amount of renewable energy sources (RESs) in power systems is of fundamental importance in reducing carbon emissions and mitigating climate change. Many regions in the United States aspire to generate 50% of their electricity from renewables by 2035, with some states targeting 100% renewables by 2050 [1]. Due to significant uncertainties and faster dynamics resulting from renewable integration and electrification, the safety margins of the electric grid have been reduced. This has been reflected by larger, more frequent, and more damaging fluctuations in both voltage and frequency. Consequently, the ability of the grid to maintain safe, efficient, and resilient operations becomes a key bottleneck for deploying these decarbonization efforts. Given that extensive infrastructure upgrades to the power system are improbable in the near to medium term because of the substantial cost of both capital and manpower, it becomes essential to deploy advanced control algorithms and operation strategies to enable a seamless transition towards a more sustainable grid.

The increased controllability of hardware, together with the explosion of available data, offers exciting opportunities to deploy advanced methods that can reshape the landscape of energy systems. For example, renewables interface with the grid through power electronic inverters, which can implement almost arbitrary control actions within their saturation limits. Properly designed control laws can offset the reduction of inertia, and lead to more resilient operations. Therefore, the bottleneck of sustainable and stable energy systems is shifting from procuring bulk physical resources to developing intelligent algorithms that more efficiently leverage these new technologies.

To make full use of the fast-responsive capacity of inverter-based RES, a number of machine learning approaches have been proposed for the controller design since conventional techniques

have difficulty coping with high nonlinearity and large dimensionality of power systems. However, for a control law to be practically applicable, provable guarantees for *large range of system parameters and typologies* are essential. A neural network that works well in training may fail to stabilize the system once implemented. Most existing learning approaches rely on finite samples and soft penalties for safety-critical constraints, which are not convincing to system operators.

In light of these challenges and opportunities, this thesis focuses on control and machine-learning methods that unlock design freedoms enabled by technology advances and provide provable guarantees on safety-critical constraints. The goal is to attain the sustainable, efficient, and safe operation of power systems.

1.2 Dissertation Outline and Contributions

This thesis develops theoretical and algorithmic frameworks for control and machine learning methods that apply to largescale and nonlinear power systems. In all of these results, structures derived from the physics of energy systems and control theory are the key to achieving safe and robust AI-based solutions. Guided by these structures, learning becomes a powerful tool for optimizing nonlinear controllers and adapting to unknown time-varying uncertainties. In turn, the results have also led to new solutions to open problems in learning-based control for general networked systems.

This thesis is outlined in two parts. In *Part I*, we show how to bridge the gap between learning and safety-critical constraints through structured neural networks guided by control theory and the physics of energy systems. The structured approach for learning-based control provides end-to-end guarantees that are independent of the learning process. In *Part II*, we further show how those end-to-end guarantees lead to more flexible learning algorithms design, including decentralized learning and trajectory generation algorithms. The remaining chapters and their contributions are elaborated as follows.

Part I: Structured Learning-Based Control with Safety-Critical Guarantees.

- Chapter 2: Reinforcement Learning for Optimal Primary Frequency Control : A Lyapunov Ap-

proach [2]. As more inverter-connected renewable resources are integrated into the grid, frequency stability may degrade because of the reduction in mechanical inertia and damping. A common approach to mitigate this degradation in performance is to use the power electronic interfaces of the renewable resources for primary frequency control. Since inverter-connected resources can realize almost arbitrary responses to frequency changes, they are not limited to reproducing the linear droop behaviors. To fully leverage their capabilities, reinforcement learning (RL) has emerged as a popular method to design nonlinear controllers to optimize a host of objective functions. Because both inverter-connected resources and synchronous generators would be a significant part of the grid in the near and intermediate future, the learned controller of the former should be stabilizing with respect to the nonlinear dynamics of the latter. To overcome this challenge, we explicitly engineer the structure of neural network-based controllers such that they guarantee system stability by construction, through the use of a Lyapunov function. A recurrent neural network architecture is used to efficiently train the controllers. The resulting controllers only use local information and outperform optimal linear droop as well as other state-of-the-art learning approaches.

- Chapter 3: Structured Neural-PI Control: Stability and Steady-State Optimality Guarantees [3, 4]. The goal is to enforce both transient stability and steady-state performances after distrubances. We propose structured neural-PI control to guarantee zero output tracking error and optimal resource allocation at the steady state. Importantly, the proposed approach achieves steady-state optimality distributedly through consensus over neighbours. This provides a systematic framework for generalized controller design with provable guarantees of stability and steady-state optimality in a wide range of networked systems.

- Chapter 4: Equilibrium-Independent Stability Analysis for Distribution Systems with Lossy Transmission Lines [5]. Because the transmission lines in distribution systems are lossy, standard approaches in power system stability analysis do not readily apply and the understanding of transient stability remains open even for simplified models. We propose a novel equilibrium-independent transient stability analysis of distribution systems with lossy lines. We certify

network-level stability by breaking the network into subsystems, and by looking at the equilibrium-independent passivity of each subsystem, the network stability is certified through a diagonal stability property of the interconnection matrix. This allows the analysis scale to large networked systems with time-varying equilibria. The proposed method gracefully extrapolates between lossless and lossy systems, and provides a simple yet effective approach to optimize control efforts with guaranteed stability regions.

Part II: Decentralized and Efficient Learning Algorithm Design.

- Chapter 5: Decentralized Safe Reinforcement Learning for Voltage Control [6, 7]. A decentralized RL framework is constructed to train local neural network controller at each bus in a model-free setting. We prove that the system is guaranteed to be exponentially stable if each controller satisfies certain Lipschitz constraints. The set of Lipschitz bound is optimized to enlarge the search space for neural network controllers. We explicitly engineer the structure of neural network controllers such that they satisfy the Lipschitz constraints by design. Simulation results show that the structure of stabilizing controllers plays a vital role for the decentralized training to converge without the need for real-time communications.
- Chapter 6: Efficient Reinforcement Learning Through Trajectory Generation [8]. A key barrier to using reinforcement learning in many real-world applications is the requirement of a large number of system interactions to learn a good control policy. To overcome these challenges, we propose a trajectory generation algorithm, which adaptively generates new trajectories as if the system is being operated and explored under the updated control policies. Motivated by the fundamental lemma for linear systems, assuming sufficient excitation, we generate trajectories from linear combinations of historical trajectories. For linear feedback control, we prove that the algorithm generates trajectories with the exact distribution as if they were sampled from the real system using the updated control policy. In particular, the algorithm extends to systems where the states are not directly observed.
- Chapter 7: Conclusions and Future Directions.

Part I

STRUCTURED LEARNING-BASED CONTROL WITH SAFETY-CRITICAL GUARANTEES

Chapter 2

REINFORCEMENT LEARNING FOR OPTIMAL FREQUENCY CONTROL: A LYAPUNOV APPROACH

2.1 Introduction

Due to the shift from conventional generation to renewable resources such as wind, solar, and storage, there has been noticeable degradation of system frequency dynamics [9]. In the near and intermediate future, both inverter-connected resources and synchronous generators would play significant roles in the grid. Therefore, the inverters still need to "play nice" with synchronous generators, where they need to respect the dynamics of the generators and help maintain the stability of the grid. A degradation in the frequency dynamics would increase the risk of load shedding and blackouts, which in turn limits the amount of renewable energy that can be integrated.

A widely adopted approach to use inverter-connected resources to provide primary frequency regulation is to engineer them to respond as conventional synchronous generators through frequency droop controls. Because of the mechanical characteristic of conventional generators, droop controls are typically linear functions of frequency deviations (with possible deadbands and saturation) [10]. Inverter-connected resources can mimic this behavior by changing their active power setpoints subject to frequency deviations [11, 12]. However, as for the common performance metrics adopted in practice, including frequency deviations and control costs [13,14], linear controllers are not optimal [15]. Since inverters are solid state electronic devices, they can implement almost arbitrary control laws by quickly adjusting their power setpoints, subject to some actuation limits [16, 17]. Then a natural question arises: *are there other control laws that still guarantee the stability of a system with synchronous generators, but have more optimal performance compared to linear droop response?*

It turns out that designing optimal controllers that respect the dynamics of power systems is

not trivial. Power system dynamics are governed by nonlinear swing equations and thus even optimizing linear controllers is a difficult problem. For nonlinear controllers, they need to be parameterized in a tractable fashion for optimization. More importantly, the controllers need to stabilize the frequency dynamics of the grid, which introduces nonlinear constraints that are not easy to work with algebraically.

A standard approach to overcome some of the above difficulties is to work with the linearized small signal model, where controllers can be designed to guarantee asymptotic stability [13, 14]. However, stability becomes more crucial when state deviations are large, where the nonlinear dynamics have to be considered. When nonlinear dynamics are considered, most approaches are restricted to tuning the slopes of the linear droop controllers [12]. To obtain better performances, model predictive control has also been used [15, 17], but they require robust real-time communication and computation capabilities, which is not yet available for much of the current system.

To break the unenviable position of not fully utilizing the capabilities of inverters for frequency control, a number reinforcement learning (RL) approaches have been proposed [18–20]. Specifically, (deep) neural networks are often used to parameterize the controllers and RL is used to train them. A number of algorithms, including deterministic policy gradient algorithm, multi-Q-learning and actor-critic methods, have been used in frequency regulation and other control problems.

The key challenge in using RL is to guarantee that learned controllers are stabilizing, that is, frequencies in the system would reach a stable equilibrium after disturbances in the system. To this end, existing approaches typically use soft penalties by adding a high cost when states leave prescribed ranges [18, 21]. However, these approaches are ad hoc. Stability should be treated as a hard constraint rather than through penalties, which is especially important since training can only be done on a limited number of samples while the controller should be stabilizing over a set of points in the state space. Another challenge comes from the controller training process. Generated trajectories are normally used to train the neural network controllers, but the evolution of state variables over long time horizons makes direct back-propagation inefficient. Approaches that use approximate value (or Q) function assume that the states are in a stationary probability distribution [22], which is generally not true during transients. Lyapunov functions have been used

as constraints [23], but learning was not considered and controllers was manually tuned.

This chapter proposes a recurrent neural network (RNN)-based RL framework to solve optimal primary frequency control problem with a stability guarantee. We derive a simple algebraic condition on the nonlinear controllers that guarantee local exponential stability of the system. More precisely, using Lyapunov theory, we show that the function from the frequency deviation to the active power output implemented by a controller needs to be monotonic and through the origin at each bus. The controllers are decentralized (each only using the frequency deviation at its own bus) and the stability guarantee holds for most system parameters and topologies.

The monotonicity of the controller is realized through a stacked-ReLU neural network which can be designed explicitly. In order to train the controllers efficiently, we design a RNN framework where the time-coupled variables in the power system form the cell component of the RNN. Simulation results show that the proposed method can learn a static nonlinear controller that performs better than traditional linear droop control. Furthermore, we show that RL without considering stability can lead to unstable controllers, whereas our approach always maintains stability. Code and data are available at https://github.com/Wenqi-Cui/RNN-RL-Frequency-Lyapunov.

2.2 Problem Setup

Consider a n-bus power system that can be modelled as a connected graph $(\mathcal{V}, \mathcal{E})$. Specifically, buses are indexed by $i, j \in \mathcal{V} := [n] := \{1, \ldots, n\}$ and transmission lines are denoted by unordered pairs $\{i, j\} \in \mathcal{E} \subset \{\{i, j\} \,|\, i, j \in \mathcal{V}, i \neq j\}$. Let states variables be phase angle $\boldsymbol{\theta} := (\theta_i, i \in [n]) \in \mathbb{R}^n$ and frequency deviation from the nominal frequency $\boldsymbol{\omega} := (\omega_i, i \in [n]) \in \mathbb{R}^n$.[1]

In this chapter, we consider static local feedback controllers: bus i measures its local frequency deviation ω_i and applies a time-invariant function to determine the control action u_i. Thus, the controller on the bus i is written as $u_i(\omega_i)$. The control action changes the *active power* coming from inverter-connected resources (e.g., solar PV, electric vehicles, and storage).

We assume the bus voltage magnitudes are 1 per unit and the reactive power flows and injections

[1]Throughout this chapter, vectors are denoted in lower case bold and matrices are denoted in upper case bold, while scalars are unbolded.

are ignored. This is the commonly used lossless power flow model, which is suitable to primary frequency control of transmission systems with small resistances and well-regulated voltages [24]. Then, the frequency dynamics is given by the swing equation [10][2]:

$$\dot{\theta}_i = \omega_i \,, \tag{2.1a}$$

$$M_i\dot{\omega}_i = p_{\mathrm{m},i} - D_i\omega_i - u_i(\omega_i) - \sum_{j=1}^{n} B_{ij}\sin(\theta_i - \theta_j) \tag{2.1b}$$

where $\boldsymbol{M} := \mathrm{diag}(M_i, i \in [n]) \in \mathbb{R}^{n\times n}$ are the generator inertia constants, $\boldsymbol{D} := \mathrm{diag}(D_i, i \in [n]) \in \mathbb{R}^{n\times n}$ are the combined frequency response coefficients from synchronous generators and frequency sensitive load, $\boldsymbol{p}_{\mathrm{m}} := (p_{\mathrm{m},i}, i \in [n]) \in \mathbb{R}^n$ are the net power injections, $\boldsymbol{B} := [B_{ij}] \in \mathbb{R}^{n\times n}$ is the susceptance matrix with $B_{ij} = 0, \forall\{i,j\} \notin \mathcal{E}$, and $\boldsymbol{u}(\boldsymbol{\omega}) := (u_i(\omega_i), i \in [n]) \in \mathbb{R}^n$.

As mentioned above, we would like to design control functions $u_i(\omega_i)$'s that can improve frequency deviation with a moderate control cost. Therefore, we consider two costs in the objective function of the optimal primary frequency control problem: the cost on frequency deviations and the cost of controllers [14, 15, 30, 31]. For a time horizon of length T, a reasonable cost on frequency deviation is represented by the infinity norm of $\omega_i(t)$ over the time horizon from 0 to T, i.e., $\|\omega_i\|_\infty := \sup_{0\le t\le T}|\omega_i(t)|$, which quantifies the maximum frequency deviation during the time horizon. For the cost on the control action, we use a quadratic cost defined by its two-norm $\|u_i\|_2^2 := \frac{1}{T}\int_0^T (u_i(t))^2 \mathrm{d}t$. The optimization problem is:

$$\min_{\boldsymbol{u}} \sum_{i=1}^{n} \left(\|\omega_i\|_\infty + \gamma\|u_i\|_2^2\right) \tag{2.2a}$$

$$\text{s.t. } \dot{\theta}_i = \omega_i \tag{2.2b}$$

$$M_i\dot{\omega}_i = p_{\mathrm{m},i} - D_i\omega_i - u_i(\omega_i) - \sum_{j=1}^{n} B_{ij}\sin(\theta_i - \theta_j) \tag{2.2c}$$

$$\underline{u}_i \le u_i(\omega_i) \le \overline{u}_i \tag{2.2d}$$

$$u_i(\omega_i) \text{ is stabilizing.} \tag{2.2e}$$

[2]The swing equations in (2.1) are called the classic second-order model. As in most existing literature [25–28], we use them in analysis. In simulations, we use 6^{nd}-order generator model with turbine-governing system as well as PLL loops on the inverters for frequency measurement [10, 29].

Here, γ in (2.2a) is a coefficient that trades off the cost of action with respect to frequency deviation. In a more general problem setting, distinct weights γ_i's can be assigned to individual control actions to achieve a desirable frequency performance at an acceptable level of control action [27]. In practice, the power inputs from inverter-based resources are always bounded by saturation. Hence, the lower and upper bounds for the control action at bus i are included as $\underline{u}_i$ and $\overline{u}_i$, respectively, in (2.2d). The special case where $\underline{u}_i = \overline{u}_i = 0$ can be used to characterize a bus i with no controllable resources. Last but not least, we include the requirement that $u_i(\omega_i)$'s stabilize the system (2.1) as a hard constraint in (2.2e).

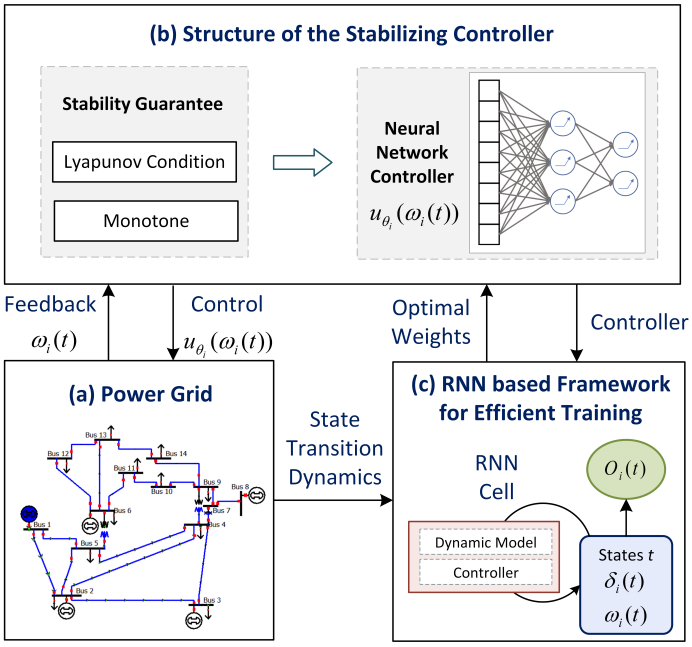


Figure 2.1: Reinforcement learning for the frequency control problem

In (2.2), we are optimizing the function $\boldsymbol{u}(\cdot)$, which is an infinite dimensional problem. To parameterize and find a good controller, reinforcement learning (RL) has emerged as an attractive alternative, where controllers are parameterized by neural networks. Thus, we parameterize each of the controllers $u_i(\omega_i)$ as a neural network with weight φ_i, sometimes written as $u_{\varphi_i}(\omega_i)$. Then, RL trains neural networks by updating φ_i's to minimize the loss given by the objective function in (2.2a).

The major challenge for RL comes from the hard constraint on the stability of the system. Although we can add a high penalty to the large magnitude of ω_i, such a penalty does not guarantee that the stability constraints are always satisfied. In fact, learned controllers that lead to reasonably looking trajectories in training may destabilize the system during testing. To overcome this challenge, we directly use the physical model (2.1) to derive the structure of the stabilizing controller based on Lyapunov stability theory. As illustrated in Fig. 2.1(b), stability can be guaranteed by enforcing a structure on the controllers $u_{\varphi_i}(\omega_i)$'s.

2.3 Structural Properties of Stablizing Controllers

To constrain the search space in (2.2) to the set of *stabilizing controllers*, we derive structural properties that the controllers should satisfy from Lyapunov stability theory. More precisely, by finding an appropriate Lyapunov function, we show that, if the output of each controller is monotonically increasing with respect to the frequency deviation, then the system has a unique equilibrium that is locally exponentially stable. In addition, we directly engineer this monotonicity feature into neural networks via properly designed weights and biases. These weights and biases are then trained to optimize the objective function in (2.2a).

2.3.1 Uniqueness of the Equilibrium

Since the frequency dynamics of the system in (2.1b) depends only on the phase angle differences, to characterize the equilibrium of the dynamics (2.1), we make the following change of coordinates:

$$\delta_i := \theta_i - \frac{1}{n}\sum_{j=1}^{n}\theta_j \,,$$

where $\boldsymbol{\delta} := (\delta_i, i \in [n]) \in \mathbb{R}^n$ can be understood as the center-of-inertia coordinates [24,32]. Then, the system dynamics in (2.1) can be written as

$$\dot{\delta}_i = \omega_i - \frac{1}{n}\sum_{j=1}^{n}\omega_j \,, \tag{2.3a}$$

$$M_i\dot{\omega}_i = p_{\mathrm{m},i} - D_i\omega_i - u_i(\omega_i) - \sum_{j=1}^{n} B_{ij}\sin(\delta_i - \delta_j) \,. \tag{2.3b}$$

Under an arbitrary control law $u_i(\omega_i)$, there may not exist a well-defined equilibrium point which the system will settle into. In the next lemma, we show that an unique equilibrium exists if the controllers satisfy a certain structure property.

Lemma 1 (Unique equilibrium). *Suppose the function $u_i(\omega_i)$ is a monotonically increasing function of the local frequency deviation ω_i. Suppose the angles satisfy $|\delta_i - \delta_j| \in [0, \pi/2)$ for all i connected to j. Then there exists an unique equilibrium point $(\boldsymbol{\delta}^*, \mathbf{1}\omega^*)$ described by*

$$0 = p_{\mathrm{m},i} - D_i\omega^* - u_i(\omega^*) - \sum_{j=1}^{n} B_{ij}\sin(\delta_i^* - \delta_j^*)\,, \tag{2.4a}$$

$$\sum_{i=1}^{n} p_{\mathrm{m},i} = \sum_{i=1}^{n} u_i(\omega^*) + \omega^* \sum_{i=1}^{n} D_i\,, \tag{2.4b}$$

if the power flow equations (2.4a) *are feasible, where* $\mathbf{1}$ *is a vector of all* 1*'s with an appropriate dimension.*

Proof. First of all, in steady state, (2.3) yields

$$0 = \omega_i^* - \frac{1}{n}\sum_{j=1}^{n} \omega_i^*\,, \tag{2.5a}$$

$$0 = p_{\mathrm{m},i} - D_i\omega_i^* - u_i(\omega_i^*) - \sum_{j=1}^{n} B_{ij}\sin(\delta_i^* - \delta_j^*)\,. \tag{2.5b}$$

Clearly, (2.5a) implies that the frequency deviation at each bus synchronizes to the same solution that $\omega_i^* = \omega^*$, and we have the desired equations in (2.4a). Since the system is lossless and $B_{ij} = B_{ji}$, the net power flow, $\sum_{i=1}^{n}\sum_{j=1}^{n} B_{ij}\sin(\delta_i^* - \delta_j^*)$, is zero. Using this fact and by summing (2.4a), we get (2.4b).

Next, we show the uniqueness of ω^* by contradiction. Suppose that both ω^* and $\omega^\star$ satisfy (2.4b), where $\omega^* \neq \omega^\star$. Then,

$$\sum_{i=1}^{n} u_i(\omega^*) + \omega^* \sum_{i=1}^{n} D_i = \sum_{i=1}^{n} u_i(\omega^\star) + \omega^\star \sum_{i=1}^{n} D_i\,,$$

which yields

$$\sum_{i=1}^{n} \frac{u_i(\omega^*) - u_i(\omega^\star)}{\omega^* - \omega^\star} = -\sum_{i=1}^{n} D_i < 0\,. \tag{2.6}$$

However, if $u_i(\omega_i)$ is monotonically increasing, the left hand side of the equality in (2.6) must be nonnegative, which is a contradiction. The uniqueness of $\boldsymbol{\delta}^*$ follows from the same argument as in [33, Lemma 1]. □

Note that the angles $\boldsymbol{\delta}$ are constrained to be in the region denoted by $\Theta := \{\boldsymbol{\delta} | |\delta_i - \delta_j| \in [0, \pi/2), \forall \{i, j\} \in \mathcal{E}\}$, which is sufficiently large to include almost all practical scenarios and is a common assumption in literature [24, 32].

2.3.2 Lyapunov Stability Analysis

In this subsection, we further show that the equilibrium point $(\boldsymbol{\delta}^*, \boldsymbol{\omega}^*)$ described by (2.4) is locally exponentially stable if the controllers are monotone. The next theorem is the main result of the paper.

Theorem 1 (Local exponential stability). *If the control output $u_i(\omega_i)$ is a monotonically increasing function of the local frequency deviation ω_i, then the equilibrium point $(\boldsymbol{\delta}^*, \mathbf{1}\omega^*)$ described by* (2.4) *is locally exponentially stable. In particular, the region of attraction include the set $\mathcal{D} := \{(\boldsymbol{\delta}, \boldsymbol{\omega}) \in \mathbb{R}^n \times \mathbb{R}^n \,|\, |\delta_i - \delta_j| \in [0, \pi/2)$ for i, j connected$\}$.*

The qualifier "local" in Theorem 1 is necessary since we need to assume that the trajectories start within the region of attraction. We note that this is far less restrictive than standard local convergence results in nonlinear systems, where the region of attraction is confined to be close to the equilibrium point [34]. The region of attraction in Theorem 1 is quite large and include most operating points of interest.

Theorem 1 gives structural properties[3] for controllers that guarantee exponential stability that does not depend on system parameter and topologies. Therefore, the optimal performance comes from training on a particular system, but the stability guarantees do not. This robustness to uncertainties is a key advantage of constraining the structure of networks compared to purely model-free RL approaches. The design of neural networks is given in the next section (Section 2.4) and the rest of this section outlines the proof of Theorem 1.

[3]These are sometimes called extended class κ functions

From Lyapunov stability theory, if there exists a Lyapunov function $V(\boldsymbol{\delta}, \boldsymbol{\omega})$ such that $\dot{V}(\boldsymbol{\delta}, \boldsymbol{\omega}) \leq -cV(\boldsymbol{\delta}, \boldsymbol{\omega})$ for a constant $c > 0$, then the system is exponentially stable [34]. Therefore, we prove Theorem 1 by constructing a qualified Lyapunov function and showing that such a constant c exist. Inspired by [32], we consider the following Lyapunov function candidate:

$$V(\boldsymbol{\delta}, \boldsymbol{\omega}) = \frac{1}{2}\sum_{i=1}^{n} M_i(\omega_i - \omega^*)^2 + W_{\mathrm{p}}(\boldsymbol{\delta}) + \epsilon W_{\mathrm{c}}(\boldsymbol{\delta}, \boldsymbol{\omega}) \tag{2.7}$$

with

$$W_{\mathrm{p}}(\boldsymbol{\delta}) := -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} B_{ij}\left(\cos(\delta_{ij}) - \cos(\delta_{ij}^*)\right) - \sum_{i=1}^{n}\sum_{j=1}^{n} B_{ij}\sin(\delta_{ij}^*)(\delta_i - \delta_i^*)\,, \tag{2.8a}$$

$$W_{\mathrm{c}}(\boldsymbol{\delta}, \boldsymbol{\omega}) := \sum_{i=1}^{n}\sum_{j=1}^{n} B_{ij}\left(\sin(\delta_{ij}) - \sin(\delta_{ij}^*)\right) M_i(\omega_i - \omega^*)\,, \tag{2.8b}$$

where $\delta_{ij} := \delta_i - \delta_j$ and $\epsilon > 0$ is a tunable parameter that should be set small enough. The physical intuition for the Lyapunov function can be found in [32,35]. Strictly speaking, this function is not a "true" Lyapunov function since it is not bounded below. The following lemma proves that $V(\boldsymbol{\delta}, \boldsymbol{\omega})$ is a well-defined Lyapunov function on the domain $\mathcal{D}$, which suffices to show that trajectories starting in $\mathcal{D}$ converge to the equilibrium. Then Lemma 3 derives the time derivative $\dot{V}(\boldsymbol{\delta}, \boldsymbol{\omega})$ and Lemma 4 shows there exists a constant $c > 0$ such that $\dot{V}(\boldsymbol{\delta}, \boldsymbol{\omega}) \leq -cV(\boldsymbol{\delta}, \boldsymbol{\omega})$.

Lemma 2 (Bounds on Lyapunov function). *$\forall(\boldsymbol{\delta}, \boldsymbol{\omega}) \in \mathcal{D}$, the Lyapunov function $V(\boldsymbol{\delta}, \boldsymbol{\omega})$ in (2.7) satisfies*

$$V(\boldsymbol{\delta}, \boldsymbol{\omega}) \geq \alpha_1(\|\boldsymbol{\delta} - \boldsymbol{\delta}^*\|_2^2 + \|\boldsymbol{\omega} - \boldsymbol{\omega}^*\|_2^2)\,,$$
$$V(\boldsymbol{\delta}, \boldsymbol{\omega}) \leq \alpha_2(\|\boldsymbol{\delta} - \boldsymbol{\delta}^*\|_2^2 + \|\boldsymbol{\omega} - \boldsymbol{\omega}^*\|_2^2)\,,$$

for some constants $\alpha_1 > 0$ and $\alpha_2 > 0$.

The proof is given in Appendix A.1. It follows directly from Lemma 2 that $V(\boldsymbol{\delta}^*, \boldsymbol{\omega}^*) = 0$ and $V(\boldsymbol{\delta}, \boldsymbol{\omega}) > 0, \forall(\boldsymbol{\delta}, \boldsymbol{\omega}) \in \mathcal{D} \setminus (\boldsymbol{\delta}^*, \boldsymbol{\omega}^*)$. To show $V(\boldsymbol{\delta}, \boldsymbol{\omega})$ is a Lyapunov function on $\mathcal{D}$, we need to show $\dot{V}(\boldsymbol{\delta}, \boldsymbol{\omega})$ decreases in $\mathcal{D}$.

Lemma 3 (Time derivative). *The time derivative of* $V(\boldsymbol{\delta}, \boldsymbol{\omega})$ *defined in* (2.7) *is given by*

$$\begin{aligned}\dot{V}(\boldsymbol{\delta}, \boldsymbol{\omega}) = &- \begin{bmatrix} \boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}) - \boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*) \\ \boldsymbol{\omega} - \boldsymbol{\omega}^* \end{bmatrix}^T \boldsymbol{Q}(\boldsymbol{\delta}) \begin{bmatrix} \boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}) - \boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*) \\ \boldsymbol{\omega} - \boldsymbol{\omega}^* \end{bmatrix} \\ &- \left[\boldsymbol{\omega} - \boldsymbol{\omega}^* + \epsilon \left(\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}) - \boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*)\right)\right]^T \left(\boldsymbol{u}(\boldsymbol{\omega}) - \boldsymbol{u}(\boldsymbol{\omega}^*)\right)\end{aligned} \tag{2.9}$$

with

$$\boldsymbol{Q}(\boldsymbol{\delta}) := \begin{bmatrix} \epsilon \boldsymbol{I} & \frac{\epsilon}{2}\boldsymbol{D} \\ \frac{\epsilon}{2}\boldsymbol{D} & \boldsymbol{D} - \frac{\epsilon}{2}(\boldsymbol{H}(\boldsymbol{\delta})\boldsymbol{M} + \boldsymbol{M}\boldsymbol{H}(\boldsymbol{\delta})) \end{bmatrix}, \tag{2.10}$$

which is positive definite for ϵ *small enough,* $\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}) := \left(p_{\mathrm{e},i}(\boldsymbol{\delta}) := \sum_{j=1}^{n} B_{ij} \sin(\delta_{ij}), i \in [n]\right) \in \mathbb{R}^n$ *and* $\boldsymbol{H}(\boldsymbol{\delta}) = \nabla \boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}) := [H_{ij}] \in \mathbb{R}^{n \times n}$ *such that*

$$H_{ij} := \begin{cases} -B_{ij}\cos(\delta_{ij}) & \text{if } i \neq j \\ \displaystyle\sum_{j'=1, j' \neq i}^{n} B_{ij'} \cos(\delta_{ij'}) & \text{if } i = j \end{cases}, \qquad \forall i, j \in [n] \ . \tag{2.11}$$

The proof is given in Appendix A.2. The cross term $\left[\boldsymbol{\omega} - \boldsymbol{\omega}^* + \epsilon \left(\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}) - \boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*)\right)\right]^T \left(\boldsymbol{u}(\boldsymbol{\omega}) - \boldsymbol{u}(\boldsymbol{\omega}^*)\right)$ generally complicates the analysis of $\dot{V}(\boldsymbol{\delta}, \boldsymbol{\omega})$. But when $u_i(\omega_i)$ is monotonically increasing with respect to ω_i, $(u_i(\omega_i) - u_i(\omega^*))$ is the same sign with $(\omega_i - \omega^*)$ and leads to nonnegative cross terms for small ϵ, implying that $\dot{V}(\boldsymbol{\delta}, \boldsymbol{\omega}) < 0, \forall (\boldsymbol{\delta}, \boldsymbol{\omega}) \in \mathcal{D} \setminus (\boldsymbol{\delta}^*, \boldsymbol{\omega}^*)$ and thus the system is locally asymptotically stable at the equilibrium point $(\boldsymbol{\delta}^*, \boldsymbol{\omega}^*)$. In the next lemma, we further show local exponential stability of the equilibrium.

Lemma 4 (Bounds on the time derivative). *If* $u_i(\omega_i)$ *is monotonically increasing with respect to* ω_i*, then there exists a constant* $\rho > 0$ *such that* $\dot{V}(\boldsymbol{\delta}, \boldsymbol{\omega}) \leq -\rho V(\boldsymbol{\delta}, \boldsymbol{\omega})$.

Proof. First, we show that the cross term related to $u_i(\omega_i)$ is nonnegative for sufficiently small ϵ. Define

$$k_i(\omega_i) := \begin{cases} \dfrac{u_i(\omega_i) - u_i(\omega_i^*)}{\omega_i - \omega_i^*} & \text{if } \omega_i \neq \omega_i^* \\ 0 & \text{if } \omega_i = \omega_i^* \end{cases}, \quad \forall i \in [n] \ .$$

Then, $\boldsymbol{K}(\boldsymbol{\omega}) := \mathrm{diag}(k_i(\omega_i), i \in \mathcal{V}) \in \mathbb{R}^{n\times n} \succeq 0$ if $u_i(\omega_i)$ is monotonically increasing with respect to ω_i. Hence,

$$\begin{aligned}
&\left[\boldsymbol{\omega}-\boldsymbol{\omega}^* + \epsilon\left(\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta})-\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*)\right)\right]^T\left(\boldsymbol{u}(\boldsymbol{\omega})-\boldsymbol{u}(\boldsymbol{\omega}^*)\right)\\
=&\left(\boldsymbol{\omega}-\boldsymbol{\omega}^*\right)^T\boldsymbol{K}(\boldsymbol{\omega})\left(\boldsymbol{\omega}-\boldsymbol{\omega}^*\right)+\epsilon\left(\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta})-\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*)\right)^T\boldsymbol{K}(\boldsymbol{\omega})\left(\boldsymbol{\omega}-\boldsymbol{\omega}^*\right)\geq 0
\end{aligned}$$

for small enough ϵ.

Then, $\dot{V}(\boldsymbol{\delta},\boldsymbol{\omega})$ can be by bounded by the quadratic term related to $\boldsymbol{Q}(\boldsymbol{\delta})$ in (2.9) as follows:

$$\begin{aligned}
\dot{V}(\boldsymbol{\delta},\boldsymbol{\omega}) \leq& -\left[\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta})-\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*)\boldsymbol{\omega}-\boldsymbol{\omega}^*\right]^T\boldsymbol{Q}(\boldsymbol{\delta})\begin{bmatrix}\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta})-\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*)\\ \boldsymbol{\omega}-\boldsymbol{\omega}^*\end{bmatrix}\\
&\overset{(a)}{\leq} -\lambda_{\min}(\boldsymbol{Q}(\boldsymbol{\delta}))\left(\|\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta})-\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*)\|_2^2+\|\boldsymbol{\omega}-\boldsymbol{\omega}^*\|_2^2\right)\\
&\overset{(b)}{\leq} -\lambda_{\min}(\boldsymbol{Q}(\boldsymbol{\delta}))\left(\gamma_1\|\boldsymbol{\delta}-\boldsymbol{\delta}^*\|_2^2+\|\boldsymbol{\omega}-\boldsymbol{\omega}^*\|_2^2\right)\\
&\leq -\lambda_{\min}(\boldsymbol{Q}(\boldsymbol{\delta}))\min(1,\gamma_1)\left(\|\boldsymbol{\delta}-\boldsymbol{\delta}^*\|_2^2+\|\boldsymbol{\omega}-\boldsymbol{\omega}^*\|_2^2\right)\\
&\overset{(c)}{\leq} -\lambda_{\min}(\boldsymbol{Q}(\boldsymbol{\delta}))\min(1,\gamma_1)\frac{1}{\alpha_2}V(\boldsymbol{\delta},\boldsymbol{\omega})\\
&\leq -\rho V(\boldsymbol{\delta},\boldsymbol{\omega})
\end{aligned} \tag{2.12}$$

with

$$\rho := \left(\min_{\boldsymbol{\delta}:|\delta_i-\delta_j|\in[0,\pi/2),\forall\{i,j\}\in\mathcal{E}}\lambda_{\min}(\boldsymbol{Q}(\boldsymbol{\delta}))\right)\frac{\min(1,\gamma_1)}{\alpha_2}>0,$$

where (a) is given by the Rayleigh-Ritz theorem, (b) is by [33, Lemma 4] with $\gamma_1 := \min_{\tilde{\delta}\in\Theta}\lambda_2(\boldsymbol{H}(\tilde{\boldsymbol{\delta}}))^2$, and (c) follows from Lemma 2. □

2.4 Design of Neural Network Controllers

In this chapter, we parametrize the controllers $u_{\varphi_i}(\omega_i)$ by a single hidden layer neural network. We assume that the processes such as automatic generation control (AGC) adjust the power setpoint of generators to make the net power injection around zero, i.e., $\sum_{i=1}^n p_{\mathrm{m},i} = 0$. For controllers $u_i(\omega_i)$'s that provide primary frequency response, we set $u_i(0) = 0$ so the controllers take no action when there is no frequency deviation. By Theorem 1, we design the neural networks to have the following structures such that the controller will be locally exponentially stabilizing:

1. $u_{\varphi_i}(\omega_i)$ is monotonically increasing;

2. $u_{\varphi_i}(\omega_i) = 0$ for $\omega_i = 0$;

3. $\underline{u}_i \leq u_{\varphi_i}(\omega_i) \leq \overline{u}_i$ (saturation constraints).

The first two requirements are equivalent to designing a monotonic increasing function through the origin. This is constructed by decomposing the function into positive and negative parts as $f_i(\omega_i) = f_i^+(\omega_i) + f_i^-(\omega_i)$, where $f_i^+(\omega_i)$ is monotonic increasing for $\omega_i > 0$ and zero when $\omega_i \leq 0$; $f_i^-(\omega_i)$ is monotonic increasing for $\omega_i < 0$ and zero when $\omega_i \geq 0$. The saturation constraints can be satisfied by hard thresholding the output of the neural network.

The function $f_i^+(\omega_i)$ and $f_i^-(\omega_i)$ are constructed using a single-layer neural network designed by stacking the ReLU function $\sigma(x) = \max(x, 0)$. Let m be the number of hidden units. For $f_i^+(\omega_i)$, let $\boldsymbol{q}_i = [q_i^1 \quad q_i^2 \quad \cdots \quad q_i^m]$ be the weight vector of bus i; $\boldsymbol{b}_i = [b_i^1 \quad b_i^2 \quad \cdots \quad b_i^m]^\intercal$ be the corresponding bias vector. For $f_i^-(\omega_i)$, let $\boldsymbol{z}_i = [z_i^1 \quad z_i^2 \quad \cdots \quad z_i^m]$ be the weights vector and $\boldsymbol{c}_i = [c_i^1 \quad c_i^2 \quad \cdots \quad c_i^m]^\intercal$ be the bias vector. Denote $\mathbf{1} \in \mathbb{R}^m$ as the all 1's column vector. The detailed construction of $f_i^+(\omega_i)$ and $f_i^-(\omega_i)$ is given in Lemma 5.

Lemma 5. *Let $\sigma(x) = \max(x, 0)$ be the ReLU function. The stacked ReLU function constructed by* (2.13) *is monotonic increasing for $\omega_i > 0$ and zero when $\omega_i \leq 0$.*

$$f_i^+(\omega_i) = \boldsymbol{q}_i \sigma(\mathbf{1}\omega_i + \boldsymbol{b}_i) \tag{2.13a}$$

$$\textit{where} \sum_{j=1}^{l} q_i^j \geq 0, \quad \forall l = 1, 2, \cdots, m \tag{2.13b}$$

$$b_i^1 = 0, b_i^l \leq b_i^{(l-1)}, \quad \forall l = 2, 3, \cdots, m \tag{2.13c}$$

The stacked ReLU function constructed by (2.14) *is monotonic increasing for $\omega_i < 0$ and zero*

when $\omega_i \geq 0$.

$$f_i^-(\omega_i) = \boldsymbol{z}_i \sigma(-\mathbf{1}\omega_i + \boldsymbol{c}_i) \tag{2.14a}$$

$$\textit{where} \sum_{j=1}^{l} z_i^j \leq 0, \quad \forall l = 1, 2, \cdots, m \tag{2.14b}$$

$$c_i^1 = 0, c_i^l \leq c_i^{(l-1)}, \quad \forall l = 2, 3, \cdots, m \tag{2.14c}$$

Proof. Note that the ReLU function $\sigma(x)$ is linear with x when activated $(x > 0)$ and equals to zero when deactivated $(x \leq 0)$, we construct the monotonic increasing function $f_i^+(\omega_i)$ by stacking the function $g_i^l(\omega_i) = q_i^l \sigma(\omega_i + b_i^l)$, as illustrated by Fig. 2.2. Since $b_i^1 = 0$ and $b_i^l \leq b_i^{(l-1)}, \forall 1 \leq l \leq m$, $g_i^l(\omega_i)$ is activated in sequence from $g_i^1(\omega_i)$ to $g_i^m(\omega_i)$ with the increase of ω_i. In this way, the stacked function is a piece-wise linear function and the slope for each piece is $\sum_{j=1}^{l} q_i^j$. Monotonic property can be satisfied as long as the slope of all the pieces are positive, i.e., $\sum_{j=1}^{l} q_i^l \geq 0, \forall 1 \leq l \leq m$. Similarly, $f_i^-(\omega_i)$ also construct by ReLU function activated for negative w_i in sequence corresponding to c_i^l for $l = 1, \cdots, m$. $\sum_{j=1}^{l} z_i^j \leq 0$ means that all the slope of the piece-wise linear function is positive and therefore guarantees monotonicity. □

Note that there still exists inequality constraints in (2.13) and (2.14), which makes the training of the neural networks cumbersome. We can reformulate the weights to get an equivalent representation that is easier to deal with in training. Define the non-negative vectors $\hat{\boldsymbol{q}}_i = \begin{bmatrix} \hat{q}_i^1 & \cdots & \hat{q}_i^m \end{bmatrix}$ and $\hat{\boldsymbol{b}}_i = \begin{bmatrix} \hat{b}_i^1 & \cdots & \hat{b}_i^m \end{bmatrix}^\intercal$. Then, (2.13b) is satisfied if $q_i^1 = \hat{q}_i^1$, $q_i^l = \hat{q}_i^l - \hat{q}_i^{(l-1)}$ for $l = 2, \cdots, m$. (2.13c) is satisfied if $b_i^1 = 0$, $b_i^l = -\sum_{j=2}^{l} \hat{b}_i^j$ for $l = 2, \cdots, m$. Similarly, define $\hat{\boldsymbol{z}}_i = \begin{bmatrix} \hat{z}_i^1 & \cdots & \hat{z}_i^m \end{bmatrix} \geq 0$ and $\hat{\boldsymbol{c}}_i = \begin{bmatrix} \hat{c}_i^1 & \cdots & \hat{c}_i^m \end{bmatrix}^\intercal \geq 0$. Then, (2.14b) is satisfied if $z_i^1 = -\hat{z}_i^1$, $z_i^l = -\hat{z}_i^l + \hat{z}_i^{(l-1)}$ for $l = 2, \cdots, m$. (2.14c) is satisfied if $c_i^1 = 0$, $c_i^l = -\sum_{j=2}^{l} \hat{c}_i^l$ for $l = 2, \cdots, m$. If the dead-band of the frequency deviation within the range $[-d, d]$ is required, it can be easily satisfied by setting $b_i^2 = -d, q_i^1 = 0$ and $c_i^2 = -d, z_i^1 = 0$ in (2.13) and (2.14).[4]

The next Theorem states the converse of Lemma 5, that is, the constructions in (2.13) and (2.14) suffice to approximate all functions of interest.

[4]A deadband is often enforced for generator droop control to reduce mechanical stress. For inverters, we do not set mandatory dead-bands.

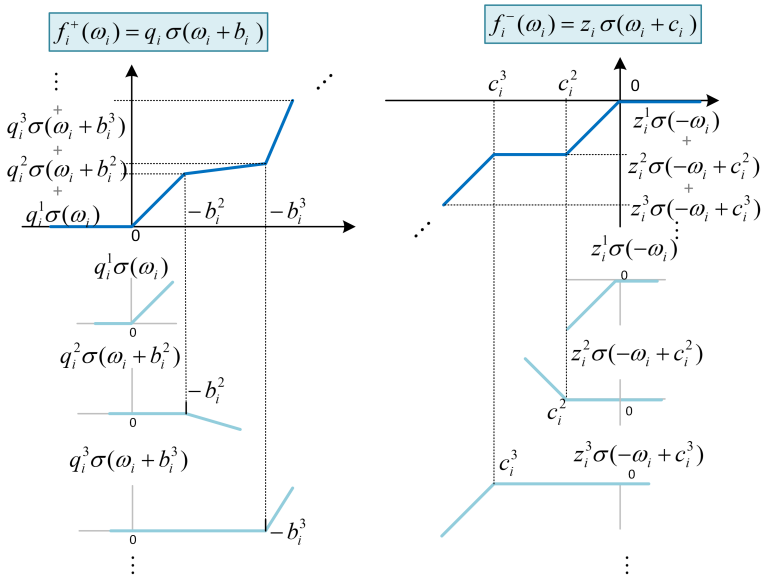


Figure 2.2: Stacked ReLU neural network to formulate a monotonic increasing function through the origin

Theorem 2. *Let $r(x)$ be any continuous, Lipschitz and bounded monotonic function through the origin with bounded derivatives, mapping compact set $\mathbb{X}$ to $\mathbb{R}$. Then there exists a function $f(x) = f^+(x) + f^-(x)$ constructed by* (2.13) *and* (2.14) *such that, for any ϵ and any $x \in \mathbb{X}$, $|r(x) - f(x)| < \epsilon$.*

The proof is given in Appendix A.3. Note that $f(x)$ is a single-layer neural network. When approximating an arbitrary function, the number of neurons and the height will depend on ϵ. Since the controller in this chapter is bounded, the stacked-ReLU neural network with limited number of neurons is sufficient for parameterization. The last step is to bound the output of the neural networks, which can be done easily using ReLU activation functions.

Lemma 6. *The neural network controller $u_i(\omega_i)$ given below is a monotonic increasing function through the origin and bounded in $[\underline{u}_i, \overline{u}_i]$ for all $i = 1, \cdots, N$:*

$$u_i(\omega_i) = \overline{u}_i - \sigma(\overline{u}_i - f_i^+(\omega_i) - f_i^-(\omega_i)) + \sigma(\underline{u}_i - f_i^+(\omega_i) - f_i^-(\omega_i)) \tag{2.15}$$

The proof of this lemma is by inspection.

2.5 Learning Control Policies Using RNNs

The structure of the controllers are decided by the constructions in (2.13), (2.14) and (2.15). In this section we develop a RNN based RL algorithm to learn their weights and biases.

Discretize Time System. To learn the controller and simulate the trajectories of the system, we discretize the dynamics (2.1) with step size Δt. We use k and K to represent the discrete time and the total number of stages, respectively. The states (θ_i, w_i) at bus i evolves along the trajectory are represented as $\boldsymbol{\theta_i} = (\theta_i(0), \theta_i(1), \cdots, \theta_i(K))$ and $\boldsymbol{\omega_i} = (\omega_i(0), \omega_i(1), \cdots, \omega_i(K))$ over K stages, with the control sequence $\boldsymbol{u_{\varphi_i}} = (u_{\varphi_i}(\omega_i(0)), \cdots, u_{\varphi_i}(\omega_i(K)))$. The infinity norm of the sequence of $\omega_i(k)$ is then defined by $||\boldsymbol{\omega_i}||_\infty = \max_{k=0,\cdots,K} |w_i(k)|$. The cost on controller is the quadratic function of action defined by its two-norm $||\boldsymbol{u_{\varphi_i}}||_2^2 = \frac{1}{K}\sum_{k=1}^{K}(u_{\varphi_i}(k))^2$. The optimization problem is

$$\min_{\boldsymbol{\varphi}} \quad \sum_{i=1}^{n} \left(||\boldsymbol{\omega_i}||_\infty + \gamma||\boldsymbol{u_{\varphi_i}}||_2^2\right) \tag{2.16a}$$

$$\text{s.t. } \theta_i(k) = \theta_i(k-1) + \omega_i(k-1)\Delta t \tag{2.16b}$$

$$\begin{aligned} \omega_i(k) = &-\frac{\Delta t}{M_i}\sum_{j=1}^{|\mathcal{B}|} B_{ij}\sin(\theta_{ij}(k-1)) + \frac{\Delta t}{M_i}p_{m,i} \\ &+ \left(1 - \frac{D_i\Delta t}{M_i}\right)\omega_i(k-1) - \frac{\Delta t}{M_i}u_{\varphi_i}(\omega_i(k-1)) \end{aligned} \tag{2.16c}$$

$$\underline{u}_i \le u_{\varphi_i}(\omega_i(k)) \le \overline{u}_i \tag{2.16d}$$

$$\omega_i(k)u_{\varphi_i}(\omega_i(k)) \ge 0 \tag{2.16e}$$

$$u_{\varphi_i}(\cdot) \text{ is increasing} \tag{2.16f}$$

and all equations hold for $i = 1, \ldots, n$. The constraints (2.16e) and (2.16f) guarantee exponentially stability.

Note that the optimization variable $\boldsymbol{\varphi}$ exists in all the time steps in (2.16). A straightforward gradient-based training approach is challenging since we need to calculate the gradient all the way to the first time step for all time steps $k = 0, \cdots, K$. To mitigate this challenge, we propose a RNN-based framework that integrates the state transition dynamics (2.16b) and (2.16c). This way,

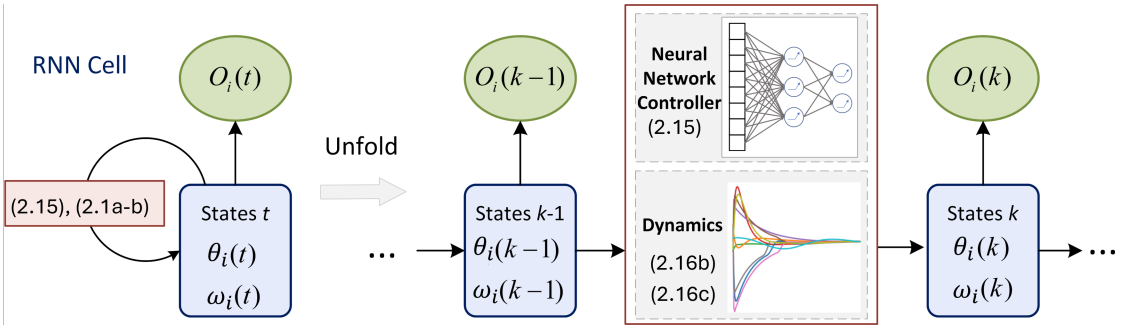


Figure 2.3: Structure of RNN for the frequency control problem

the gradient of the optimization objective with respect to φ can be computed efficiently through back-propagation.

RNN for control. RNN is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. By defining the cell state as the time-coupled states θ_i and ω_i, the state transition dynamics of the power system is integrated as illustrated in Fig. 2.3

The operation of RNN is shown by the left side of Fig. 2.3. The cell unit of RNN will remember its current state at the stage k and pass it as an input to the next stage. Unfolding the cell unit through time will give the right side of Fig. 2.3. In this way, RNN can be utilized to deal with time-coupled state variables. Specifically, the state $(\theta_i(k-1), \omega_i(k-1))$ for all $i = 1, \cdots, n$ at the stage $k-1$ is taken as an input in the state transition function (2.16b) (2.16c) and thus the state $(\theta_i(k), \omega_i(k))$ for all $i = 1, \cdots, n$ at the stage k is obtained. The control function $u_{\varphi_i}(\omega_i(k))$ in the state transition function is formatted through (2.15) to satisfy inequality constraints. The output $O_i(k) = [O_i^1(k) \quad O_i^2(k)]$ at stage k is a vector with two components computed by $O_i^1(k) = \omega_i(k)$ and $O_i^2(k) = (u_{\varphi_i}(\omega_i(k)))^2$. The loss function is formulated to be equivalent with the objective function (2.16a) as:

$$Loss = \sum_{i=1}^{N} \max_{k=0,\cdots,K} |O_i^1(k)| + \gamma \frac{1}{K} \sum_{k=1}^{K} O_i^2(k) \tag{2.17}$$

The trainable variables φ is specified in the neural network controller (2.15) and updated by gra-

dient descent through the Loss function (2.17). The unfolded structure of RNN form a directed graph along a temporal sequence where the gradient of Loss function can be efficiently computed by auto-differentiation mechanisms [36].

Algorithm. The pseudo-code for our proposed method is given in Algorithm 1. The variables to be trained are weights $\boldsymbol{\varphi} = \{\hat{\boldsymbol{q}}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{z}}, \hat{\boldsymbol{c}}\}$ for control network represented by (2.13)-(2.15) . The $i-th$ row of $\hat{\boldsymbol{q}}$ and $\hat{\boldsymbol{z}}$ are the vector $\hat{\boldsymbol{q}}_i$ and $\hat{\boldsymbol{z}}_i$ in (2.13) and (2.14), respectively. The i-th column of $\hat{\boldsymbol{b}}$ and $\hat{\boldsymbol{c}}$ are the vector $\hat{\boldsymbol{b}}_i$ and $\hat{\boldsymbol{c}}_i$ in (2.13) and (2.14), respectively. Training is implemented in a batch updating style where the h-th batch initialized with randomly generated initial states $\{\theta_i^h(0), \omega_i^h(0)\}$ for all $i = 1, \cdots, n$. The evolution of states in K stages will be computed through the structure of RNN as shown by Fig. 2.3. Adam algorithm is adopted to update weights in each episode.

2.6 Experimental Results

Case studies are conducted on the IEEE New England 10-machine 39-bus (NE39) power networks to illustrate the effectiveness of the proposed method. Firstly, we show that the proposed Lyapunov-based approaches for designing neural network controller can guarantee stability, while unconstrained neural networks may result in unstable controllers. Then, we show that the proposed structure can learn a nonlinear controller that performs better than other controllers. To ensure that our results apply in practice, simulations are conducted on the system with 6^{nd}-order generator model as well as PLL loops on the inverters for frequency measurement [29, 37].

Simulation Setting. We use TensorFlow 2.0 framework to build the reinforcement learning environment and run the training process in Google Colab with a single Nvidia Tesla P100 GPU with 16GB memory. Power System Toolbox (PST) in MATLAB is utilized to simulate the dynamic response from 6-order generator model with turbine-governing system and 2-order phase-locked-loop (PLL) block on the inverter-connected resources [29, 37]. Parameters for the transient and sub-transient process of generators are obtained in [38]. The system is in the Kron reduced form [15, 39] and its dynamics is represented by (2.1). The bound on action $\overline{u}_i$ is generated to be uniformly distributed in $[0.8P_i, P_i]$. The initial states of angle and frequency are randomly gener-

Algorithm 1: Reinforcement Learning with RNN

Require: Learning rate α, batch size H, total time stages K, number of episodes I, parameters in optimal frequency control problem (2.16)

Input: The bound of $\overline{\theta}_i$ and $\overline{\omega}_i$ to generate the initial states

Initialisation :Initial weights $\boldsymbol{\varphi}$ for control network

for $episode = 1$ *to* I **do**

  Generate initial states $\theta_i^h(0), \omega_i^h(0)$ for the i-th bus in the h-th batch, $i = 1, \cdots, n$, $h = 1, \cdots, H$;

  Reset the state of cells in each batch as the initial value $x_i^h \leftarrow \{\theta_i^h(0), \omega_i^h(0)\}$.;

  RNN cells compute through K stages to obtain the output $\{O_{h,i}(0), O_{h,i}(1), \cdots, O_{h,i}(K)\}$;

  Calculate total loss of all the batches $Loss = \frac{1}{H}\sum_{h=1}^{H}\sum_{i=1}^{N}\max_{k=0,\cdots,K}|O_{h,i}^1(k)| + \gamma\frac{1}{K}\sum_{k=1}^{K}O_{h,i}^2(k)$;

  Update weights in the neural network by passing $Loss$ to Adam optimizer: $\boldsymbol{\varphi} \leftarrow \boldsymbol{\varphi} - \alpha\text{Adam}(Loss)$

end

ated such that $\delta_i(0)$ is uniformly distributed in $[-0.05, 0.05]$ rad, $\omega_i(0)$ is uniformly distributed in $[-0.1, 0.1]$ Hz. The cost coefficient $\gamma = 0.01$. The stepsize between time states is set as $\Delta t = 0.01s$ and the total time stages is $K = 200$.

We compare the performance of the proposed RNN based structure where the neural network controller is designed with and without the Lyapunov-based approach, and the drop control with optimized linear coefficient. The parameter settings are as follows:

1. RNN-Lyapunov: Neural network controller designed based on Algorithm 1, which satisfies Theorem 1. The episode number, batch size and the number of neurons are 600, 800 and 20, respectively. Parameters of RNN are updated using Adam with learning rate initializes at 0.05 and decays every 30 steps with a base of 0.7.

2. RNN-Wo-Lyapunov: Controllers are learned without imposing any structures and purely optimizes the reward during training. The controllers are parametrized as neural networks with two dense-layer and the activation function in the first layer is $\tanh$. All the other parameters are the same as RNN-Lyapunov.

3. Linear droop control: let k_i be the droop coefficient for bus i and the droop control policy is $u_i(\omega_i) = k_i\omega_i$ for $i = 1, \ldots, n$, thresholded to their upper and lower bounds. The optimized droop coefficient is obtained by fmincon function of Matlab.

4. PG-Monotone: This controller is to demonstrate the performance improvements of using RNN during training. So here we impose the stacked-ReLU structure and trained with REINFORCE Policy Gradient algorithm [22]. The neural networks for controller, the episode number, batch number and optimizer are the same as RNN-Lyapunov. The learning rate initializes at 0.01 and decays every 30 steps with a base of 0.7.

Necessity of Lyapunov-based Approach. Theorem 1 ensures that the learned controller would be locally exponentially stable, but it's interesting to check the performance of an unconstrained controller. Intuitively, an unstable controller should lead to large costs since some trajectories would be blowing up. Then maybe a controller that minimizes the cost would also be stabilizing.

Figure 2.4 shows the training loss between controllers learned with and without the Lyapunov-based approach. Both losses converge, with the Lyapunov-based controller having better performances. However, when we implement the controllers, the one without considering stability is unstable and leads to very large state oscillations (Fig. 2.5b). In contrast, the controller constrained by the Lyapunov condition shows good performance (Fig. 2.5a). The reason for this dichotomy in performance is that we can only check a finite number of trajectories during training, and good training performance does not in itself guarantee good generalization. Therefore, explicitly constraining the controller structure is necessary.

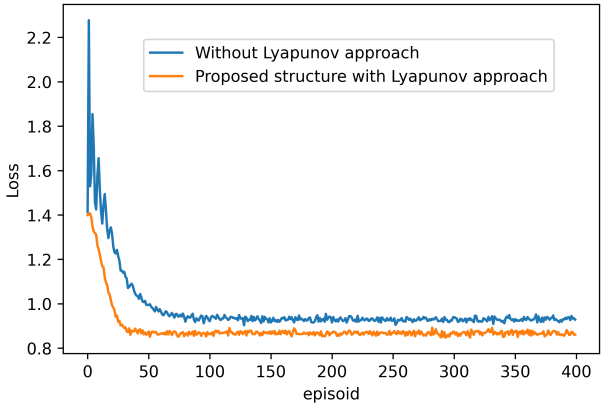
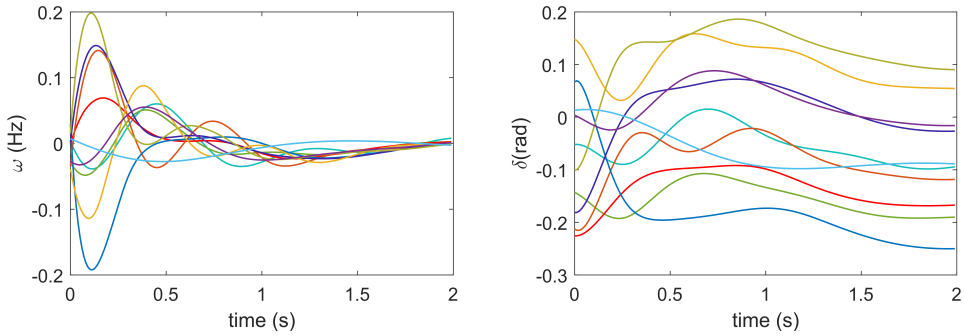
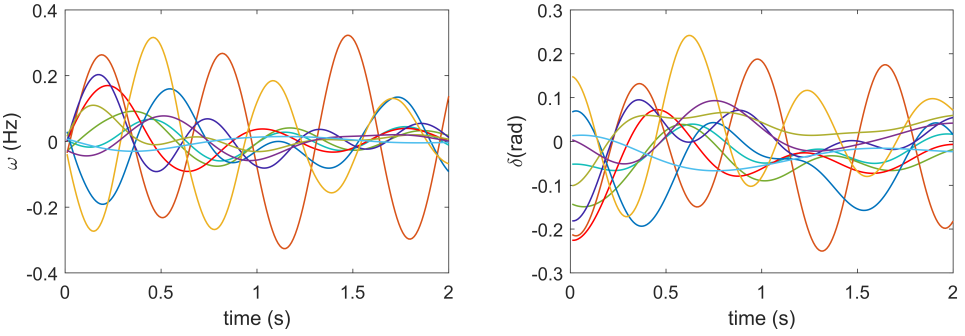


Figure 2.4: Average batch loss along episodes for controller designed with and without the Lyapunov approach. Both converges, with the former converging much for quickly than the latter.



(a) Dynamics of ω (left) and δ (right) for RNN-Lyapunov



(b) Dynamics of ω (left) and δ (right) for RNN-Wo-Lyapunov

Figure 2.5: Dynamics of angle δ and frequency deviation ω in 10 generator buses corresponding to (a) the neural network controller designed with the Lyapunov-based approach and (b) the neural network controller designed without the Lyapunov-based approach. The two controllers exhibit qualitatively different behavior even though they both achieve finite training losses in Fig. 2.4. The controller designed without the Lyapunov approach leads to unstable trajectories of the system.

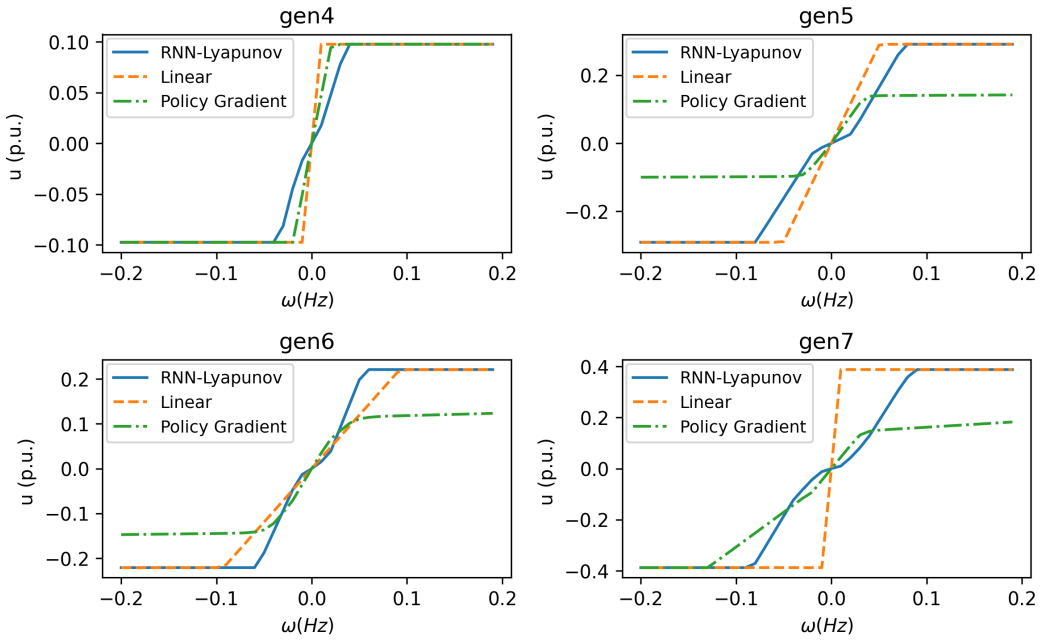


Figure 2.6: Examples of learned controller u corresponding to RNN-Lyapunov, Linear droop control and Policy Gradient for generator buses 4,5,6 and 7. The comparison shows that the proposed Stacked-ReLU neural network learns nonlinear controllers in flexible shapes.

Performance Comparisons. This subsection shows that the proposed method can learn a static nonlinear controller that outperforms the optimal linear droop controller and the RNN training technique is much more efficient than using a standard policy gradient method. Figure 2.6 illustrates the control policy learnt from RNN-Lyapunov, Policy Gradient and the linear droop control with optimized droop coefficient for four generators. Compared with the traditional droop control, the proposed stacked-ReLU neural network learns a nonlinear controller with different shapes for RNN-Lyapunov and PG-Monotone.

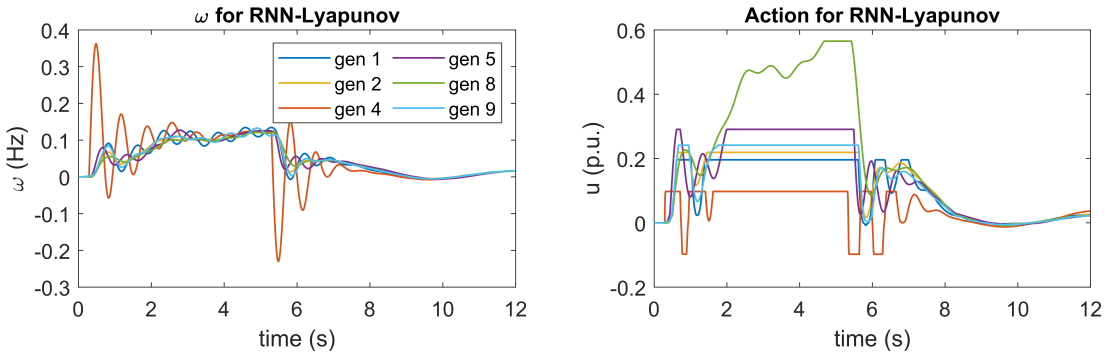
We first study the learned controllers and their performances during a sudden change in load or generation. Suppose bus 4 experiences a step load increase of 0.05 p.u. occurs at t=0.3s and a step load recovery occurs at t=5.3s. Figure 2.7 illustrates the dynamics of ω and corresponding control action u under each of the controllers. After the step load change, RNN-Lyapunov and linear droop control achieve similar maximum frequency deviation, while the control action of

RNN-Lyapunov is much lower than the other. PG-Monotone shows higher frequency deviation and oscillations. Therefore, the proposed RNN-Lyapunov approach has the minimal cost. The computational time of the proposed RNN based method is 1080.38s, while the computational time of REINFORCE policy gradient takes 4206.43s. Therefore, the proposed RNN based structure reduces computational time by approximate 74.32% compared with the general RL structure.

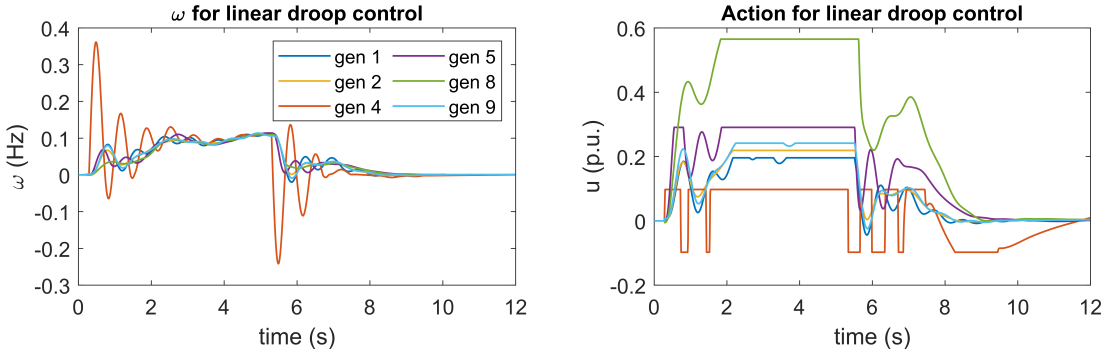
Next, we randomize the initial starting points to simulate and test the performance of the three methods under multiple different trajectories. We fix initial δ in $U[-0.1, 0.1]$ rad and let the initial ω to uniformly distributed in $U[-\bar{\omega}, \bar{\omega}]$ around the equilibrium, where $\bar{\omega}$ denotes the variation bound of initial ω. The average loss corresponding to $\bar{\omega} = 0.0, 0.025, \cdots, 0.15$ Hz are illustrated in Fig. 2.8. $\bar{\omega} = 0$ is the case that no variation of ω exists in the initial condition. Overall, RNN-Lyapunov remains approximate 11.39% and 5.41% lower in average loss than that of linear droop control, PG-Monotone, respectively. Therefore, the proposed method learn the nonlinear controller that leads to better average control performance under different initial conditions.

2.7 Conclusion

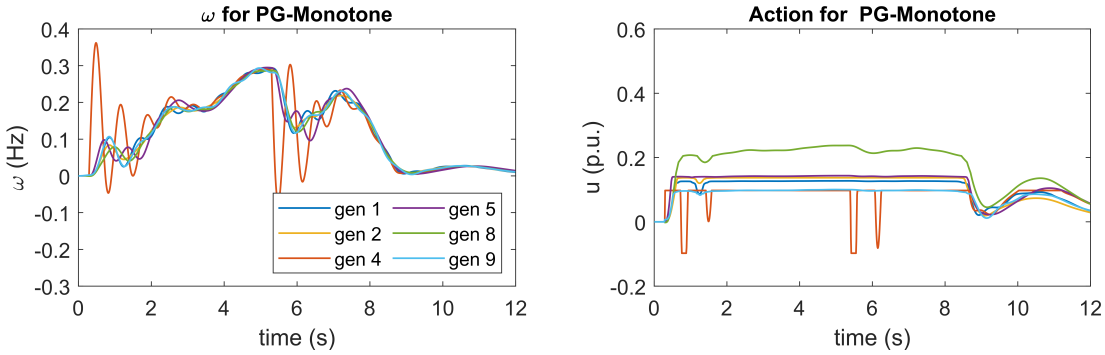
This chapter investigates the optimal frequency control problem using reinforcement learning with stability guarantees. From Lyapunov stability theory, We construct the controllers to be monotonically increasing through the origin, and prove they guarantee stability for all operating points in a region. These controllers are trained using a RNN-based method that allows for efficient back propagation through time. The learned controllers are static piece-wise linear functions that do not need real-time computation and is practical for implementation. Through simulations, we show that they outperform optimal linear droop as well as purely unstructured controllers trained via reinforcement learning. In particular, controllers failing to consider stability constraints in learning may lead to unstable trajectories of the state variables, while our proposed controllers can achieve optimal performances in system frequency responses that use small control efforts.



(a) Dynamics of ω (left) and u (right) for RNN-Lyapunov



(b) Dynamics of ω (left) and u (right) for linear droop control



(c) Dynamics of ω (left) and u (right) for controller obtained by PG-Monotone

Figure 2.7: Dynamics of the frequency deviation w and the control action u in selected generator buses corresponding to (a) Lyapunov-guided neural network controller learned with RNN. (b) Linear droop control. The proposed RNN controller has the smallest cost.

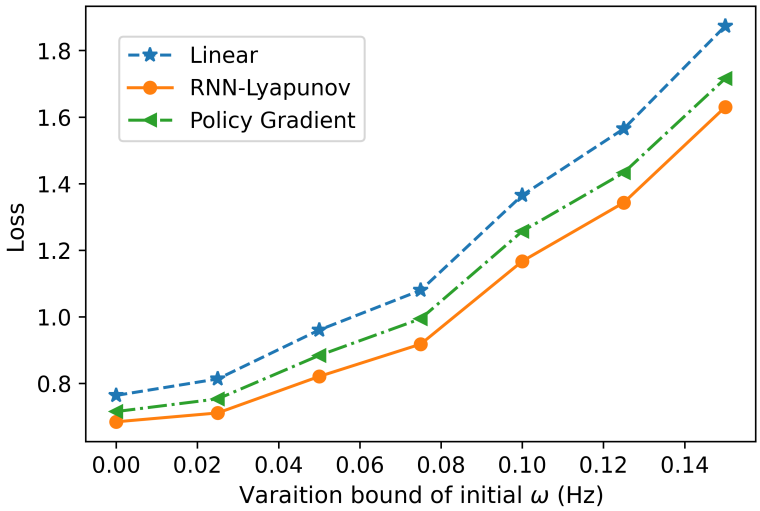


Figure 2.8: Loss with different variation range of initial conditions for RNN-Lyapunov, Linear droop controller and Policy Gradient. Compared with Linear droop controller and PG-Monotone, RNN-Lyapunov reduces the loss by approximate 11.39%, 5.41%, respectively.

Chapter 3

STRUCTURED NEURAL-PI CONTROL: STABILITY AND STEADY-STATE OPTIMALITY GUARANTEES

3.1 Introduction

The key to the normal operation of a power system is the balance between electric power supply and demand over the network [40]. For instance, the main cause of the 2021 Texas power crisis is that the deficient supply of power due to frozen equipment could not meet the high demand for electricity in cold weather. A system frequency deviation from its nominal value is a reflection of a power imbalance [41], which makes frequency control a vital task of grid operators. Traditionally, this task is performed in a hierarchical structure composed of three layers with timescale separation: primary—droop control ($<20\,\mathrm{s}$), secondary—frequency restoration ($30\,\mathrm{s}$–$10\,\mathrm{min}$), and tertiary—economic dispatch ($>15\,\mathrm{min}$) [41].

Nowadays, power systems are experiencing a change in the mix of generation, where conventional synchronous generators are gradually being replaced by renewable energy sources like solar and wind energy [42]. It is anticipated that the renewable share of the electricity generation mix in the United States will double from 21% in 2020 to 42% in 2050 [43]. Intermittent renewable sources are typically inverter-interfaced, which may adversely affect the robustness of the frequency dynamics due to the loss of inertia [44]. This exposes power systems to larger and faster frequency fluctuations than before, which has motivated active research on flexible distributed frequency control schemes that can break the hierarchy by addressing simultaneously frequency degradation and economic efficiency at a fast timescale.

A key challenge in frequency control is that the power imbalances across the network are not explicitly known. A number of studies have proposed distributed algorithms to overcome this challenge. The works in [14, 33, 45–51] focus on optimizing the steady-state frequency and economic

performance using a principled design of fixed updating rules involving agent communication for control dynamics. The approaches mainly fall in two categories. The first category [14, 46, 49–51] rests on a primal-dual interpretation of power system dynamics under a properly designed optimization problem. This approach, however, always requires the estimation of certain system parameters. The second category [33, 45, 47, 48] builds upon various consensus algorithms to converge to an equilibrium with nominal frequency and economic efficiency. A notable example in this category is the distributed averaging-based integral (DAI) mechanism [33, 45, 48, 52], where the controllable power injections are directly proportional to the integrals of frequency deviation and economic inefficiency signals. A key caveat to this approach is that DAI control has been so far restricted to quadratic cost functions.

The works above focus on the optimization of the steady-state performance and do not typically consider the transient performance along the system trajectories following power disturbances. In fact, the optimization of transient performance is a challenging problem due to the nonlinearity of power dynamics and the uncertainty in power disturbances. Reinforcement learning (RL) [53–61] is a powerful tool for learning from interactions with uncertain environments and determining how to map situations to actions so that a desired performance is optimized. By virtue of the above feature, RL has emerged [55, 56] as an effective instrument to address the optimal transient frequency control problem in nonlinear power systems under unknown power disturbances. Nevertheless, the Achilles' heel of standard RL algorithms is their lack of provable stability guarantees, which presents a significant barrier to their practical implementation for the operation of power systems. In fact, many works [57–61] optimize the transient performance by learning control policies that exhibit good performance against data but without any provable guarantees on steady-state performance. The work in Chatper 2 proposes a way to address the stability issue by identifying a set of properties that make a control design stabilizing and then restricting the search space of neural network-based controllers. In this chapter, we extend this idea to achieve provable guarantees on frequency restoration and economic efficiency at the steady-state.

With the aim of filling the gap between the optimization of steady-state and transient frequency control performance, We propose a structured neural-PI controller that has provable stability guar-

antees and achieves steady-state economic dispatch. The key structure we use are monotonically increasing functions, and they are parameterized by what we call monotone neural networks. This way, transient performances can be optimized by the training of monotone neural networks, while stability and steady-state optimality are inherently guaranteed by design. Experiments demonstrate that the proposed approach can reduce the transient cost by at least 30% compared to optimized linear controllers, ensure stability and obtain optimal steady-state cost when communication is available. Unstructured neural networks, on the other hand, often lead to unstable behaviors.

3.2 Problem Statement

We consider the power system model the same as Section 2.2. Since the frequency dynamics depend only on the phase angle differences, for the convenience of analysis, we express the dynamics the same as (2.3) in center-of-inertia coordinates [32, 33]. Observe from (2.5) that, with purely primary frequency control, the system undergoing power disturbances $\boldsymbol{p}$ may synchronize to a nonzero frequency deviation, i.e., $\omega^* \neq 0$, since $\omega^* = 0$ only if $\sum_{i=1}^n p_i + \sum_{i=1}^n u_i^* = 0$. Therefore, we hope to regulate the frequency such that $\omega^* = 0$ by providing appropriate controllable power injections $\boldsymbol{u}$ to meet power disturbances in $\boldsymbol{p}_{\mathrm{m}}$.

3.2.1 Performance Assessment

For the design of frequency control strategies, not only frequency performance but also economic factors must be taken into account. Moreover, the secure and efficient operation of power systems relies on properly controlled frequency and cost in both slow and fast timescales. Thus, we now introduce the frequency and economic performance metrics used in this manuscript based on different timescales.

Steady-State Performance Metrics. Our control objective in the long run is to achieve the nominal frequency restoration, i.e., $\boldsymbol{\omega}^* = \mathbb{0}_n$, as well as the lowest steady-state aggregate operational cost $C(\boldsymbol{u}^*) := \sum_{i=1}^n C_i(u_i^*)$, where the cost function $C_i(u_i)$ quantifies either the generation cost on a generator bus or the user disutility on a load bus for contributing u_i. This results in the fol-

lowing constrained optimization problem called *optimal steady-state economic dispatch problem*:

$$\min_{\boldsymbol{u}^*} \quad C(\boldsymbol{u}^*) := \sum_{i=1}^{n} C_i(u_i^*) \tag{3.1a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} p_i + \sum_{i=1}^{n} u_i^* = 0\,, \tag{3.1b}$$

where (3.1b) is a necessary constraint on the steady-state controllable power injections $\boldsymbol{u}^*$ in order to achieve $\boldsymbol{\omega}^* = \mathbb{0}_n$. Here, we adopt the standard assumption that the cost function $C_i(u_i)$ is strictly convex and continuously differentiable [47, 50] with respect to u_i. Then, the optimization problem (3.1) has a convex objective function and an affine equality constraint, which gives the following lemma.

Lemma 7 (Equivalent condition for economic dispatch). *Suppose the cost function $C_i(u_i)$ is strictly convex and continuously differentiable with respect to u_i, then $\boldsymbol{u}^*$ is the unique minimizer of* (3.1) *if and only if it ensures* identical marginal costs*, i.e.,*

$$\nabla C_i(u_i^*) = \nabla C_j(u_j^*)\,, \qquad \forall i, j \in \mathcal{N}\,. \tag{3.2}$$

The proof follows the Karush-Kuhn-Tucker conditions [62, Chapter 5.5.3] and is given in [32, 45, 47, 48, 52].

Transient Performance Metrics. Following sudden major power disturbances, the transient frequency dip can be large in the first few seconds, especially in low-inertia power systems. This may trigger undesired protection measures and even cause cascading failures. Thus, besides the steady-state performance, one should also pay attention to the transient frequency performance with moderate economic cost. With this aim, we define the following transient performance metrics evaluated along the trajectories of the system:

- *Frequency Nadir* is the maximum frequency deviation from the nominal frequency on each bus during the transient response, i.e.,

$$\|\omega_i\|_\infty := \max_{t\geq 0} |\omega_i(t)|\,. \tag{3.3}$$

- *Finite horizon economic cost* measures the average cost on a generator or load bus for its participation in frequency control during a time horizon T, i.e.,

$$\bar{C}_{i,T} := \frac{1}{T}\int_0^T C_i(u_i(t))\mathrm{d}t\,.$$

Then the *optimal transient frequency control problem* becomes:

$$\min_{\boldsymbol{u}} \quad \sum_{i\in\mathcal{G}} \|\omega_i\|_\infty + \rho\sum_{i=1}^{n}\bar{C}_{i,T} \quad \text{s.t.} \quad \boldsymbol{u} \text{ stabilize (2.3)}\,, \tag{3.4}$$

where $\rho > 0$ is the coefficient for tradeoff between the frequency performance and the economic cost. Note that we impose the stability requirement on $\boldsymbol{u}$ as a hard constraint in the optimization problem (3.4), which will play a pivotal role in its design.

Our goal is to design an optimal stabilizing controller that brings the system to an equilibrium that restores the nominal frequency, i.e., $\boldsymbol{\omega}^* = \mathbb{0}_n$, and solves the optimal steady-state economic dispatch problem (3.1), while solving the optimal transient frequency control problem (3.4) at the same time. A good starting point to achieve our steady-state control goal is the well-known DAI control. However, it is not straightforward how to also optimize the transient performance. This is hard to be done purely by conventional optimization methods since power systems are nonlinear and power disturbances are unknown. Therefore, we would like to integrate RL into DAI to jointly optimize steady-state and transient performance.

3.3 Generalized Proportional-Integral Control

3.3.1 Economic dispatch at the steady-state

By (3.2), enforcing that $\boldsymbol{u}^*$ achieves identical marginal cost, i.e., $\nabla C_i(u_i^*) = \nabla C_j(u_j^*), \forall i,j \in \mathcal{V}$ can ensure that the steady-state actions settle down to the solution of the resource allocation problem. To this end, prior works have designed distributed averaging-based integral control by communicating $\nabla C_i(u_i)$ with its neighbours [33, 45, 63]. However, they are restricted to quadratic costs and linear controllers. In this chapter, we consider nonlinear controllers and a more general class of cost functions in the following assumption.

Assumption 1 (Scaled-cost gradient functions). *The function $C_i(\cdot) : \mathbb{R} \mapsto \mathbb{R}$ is strictly convex and continuously differentiable for all $i \in \mathcal{V}$. Moreover, there exists a function $C_{\mathrm{o}}(\cdot) : \mathbb{R} \mapsto \mathbb{R}$ and a group of positive scaling factors $\boldsymbol{c} := (c_i, i \in \mathcal{V})$ such that $\nabla C_i(\cdot) = \nabla C_{\mathrm{o}}(c_i \cdot), \forall i \in \mathcal{V}$.*

Some examples satisfying Assumption 1 are 1) *polynomials of the form:* $C_i(u_i) = \frac{c_i}{p} u_i^p + b_i$ where $c_i > 0$ and p is an even integer (this includes quadratics). 2) *functions that are identical up to constants:* $C_i(u_i) = C_{\mathrm{o}}(u_i) + b_i$ (e.g., power generators of the same type but with different startup costs).

3.3.2 Structured controller design

We aim to design the control law such that the control effort reaches the solution of the economic dispatch problem (3.1) at the steady state, which can be equivalently realized through identical marginal cost at the steady state by Lemma 7. Hence, we design the mechanism such that neighbouring nodes communicate their marginal cost and reach the consensus at the steady state. We model communication network within the physical networked system as a connected graph $\tilde{\mathcal{G}} = (\mathcal{V}, \tilde{\mathcal{E}})$ with an incidence matrix $\tilde{\boldsymbol{E}}$. By adding the communication loop into the integral variable $\boldsymbol{s}$, the integral control term $\boldsymbol{\pi}^I(\boldsymbol{s})$ can respond to the difference of marginal costs. The edges $l = (i, j) \in \tilde{\mathcal{E}}$ are not necessarily the same as $\mathcal{E}$ and we use $\tilde{}$ to denote all variables belonging to the edges in the communication graph. The communication network associated with nodes of the physical network is designed as follows

$$\dot{s}_i = \omega_i - c_i \sum_{l=1}^{m} \tilde{\boldsymbol{E}}_{i,l} \phi_l \left(\nabla C_i(u_i(s_i)) - \nabla C_j(u_j(s_i)) \right) \tag{3.5}$$

Compactly, we have the closed-loop dynamics for the communication graph represented by $\dot{\boldsymbol{s}} = \boldsymbol{\omega} - \hat{\boldsymbol{c}} \tilde{\boldsymbol{E}} \boldsymbol{\phi} \left(\tilde{\boldsymbol{E}}^\top \nabla \boldsymbol{C}(\boldsymbol{\pi}^I(\boldsymbol{s})) \right)$, where $\hat{\boldsymbol{c}} = \mathrm{diag}(c_1, \cdots, c_n)$. Then, the control law is designed as follows.

Controller Design 1 (Distributed Steady-State Optimization). *For each node $i \in \mathcal{V}$, the control law is $u_i = \pi_i^P(\omega_i) + \pi_i^I(s_i)$, where $\pi_i^P(\cdot) : \mathbb{R} \mapsto \mathbb{R}$ and $\pi_i^I(\cdot) : \mathbb{R} \mapsto \mathbb{R}$ are Lipschitz continuous and strictly increasing functions with $\pi_i^P(0) = 0$, $\pi_i^I(0) = 0$. The ancillary state $\boldsymbol{s}$ comes from the*

communication network (3.5) *where the function* $\phi_l(z) : \mathbb{R} \mapsto \mathbb{R}$ *is an odd function and with the same sign as* z *for all* $l \in \tilde{\mathcal{E}}$*. Compactly, we have*

$$\boldsymbol{u} = \boldsymbol{\pi}^P(\boldsymbol{\omega}) + \boldsymbol{\pi}^I(\boldsymbol{s}) \tag{3.6a}$$

$$\dot{\boldsymbol{s}} = \boldsymbol{\omega} - \underbrace{\hat{\boldsymbol{c}}\tilde{\boldsymbol{E}}\boldsymbol{\phi}\left(\tilde{\boldsymbol{E}}^\top \nabla \boldsymbol{C}(\boldsymbol{\pi}^I(\boldsymbol{s}))\right)}_{\textit{the term with communication}} \tag{3.6b}$$

The following lemma shows properties of the added term in (3.6), providing an intuition about why Controller Design 1 can guarantee identical marginal cost (i.e., $\nabla C(\boldsymbol{r}(\boldsymbol{s}^*)) \in \mathrm{range}(\mathbb{1}_n)$).

Lemma 8 (Cross term in the communication network). *Suppose Assumption 1 holds. Then*

$$\hat{\boldsymbol{c}}\tilde{\boldsymbol{E}}\boldsymbol{\phi}\left(\tilde{\boldsymbol{E}}^\top \nabla \boldsymbol{C}\left(\boldsymbol{\pi}^I\left(\boldsymbol{s}\right)\right)\right) = \mathbb{0}_n \tag{3.7}$$

if and only if $\nabla \boldsymbol{C}(\boldsymbol{\pi}^I(\boldsymbol{s})) \in \mathrm{range}(\mathbb{1}_n)$*. Moreover,* $\boldsymbol{\pi}^I(\boldsymbol{s})^T \hat{\boldsymbol{c}}\tilde{\boldsymbol{E}}\boldsymbol{\phi}\left(\tilde{\boldsymbol{E}}^\top \nabla \boldsymbol{C}\left(\boldsymbol{\pi}^I\left(\boldsymbol{s}\right)\right)\right) \geq 0$ *with equality holds if and only if* $\nabla \boldsymbol{C}(\boldsymbol{\pi}^I(\boldsymbol{s})) \in \mathrm{range}(\mathbb{1}_n)$.

The proof is given in Appendix B.1 by expanding the terms and the properties of cost functions satisfying Assumption 1. In particular, we use the fact that $\phi_l(\cdot)$ is an odd function. We will show in the next subsection that $\boldsymbol{\omega}^* = \mathbb{0}_n$ is maintained and thus $\boldsymbol{u}^* = \boldsymbol{\pi}^I(\boldsymbol{s}^*)$. Then, $\nabla \boldsymbol{C}(\boldsymbol{\pi}^I(\boldsymbol{s}^*)) \in \mathrm{range}(\mathbb{1}_n)$ is equivalent to $\nabla \boldsymbol{C}(\boldsymbol{u}^*) \in \mathrm{range}(\mathbb{1}_n)$.

3.3.3 Unique equilibrium with steady-state optimality

The next theorem states that the closed-loop system (2.3) with Controller Design 1 yields a unique equilibrium that guarantees output agreement and optimal resources allocation at the steady state.

Theorem 3 (Steady-state optimality). *Suppose Assumption 1 hold and* $\forall\{i,j\} \in \mathcal{E}$*,* $|\delta_i^* - \delta_j^*| \in [0, \pi/2)$*. The closed-loop system* (2.3) *with* $\boldsymbol{u}$ *following* (3.6) *has an unique equilibrium characterized by*

$$\boldsymbol{\omega}^* = \mathbb{0}_n, \tag{3.8a}$$

$$\boldsymbol{u}\left(\boldsymbol{s}^*\right) = \nabla C_{\mathrm{o}}^{-1}(\gamma)\hat{\boldsymbol{c}}^{-1}\mathbb{1}_n\,, \tag{3.8b}$$

$$\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}) = \boldsymbol{p}_{\mathrm{m}} - \nabla C_{\mathrm{o}}^{-1}(\gamma)\hat{\boldsymbol{c}}^{-1}\mathbb{1}_n\,, \tag{3.8c}$$

where $\nabla C_{\mathrm{o}}^{-1}(\cdot)$ *is the inverse of* $\nabla C_{\mathrm{o}}(\cdot)$ *and* γ *is the unique solution to*

$$\nabla C_{\mathrm{o}}^{-1}(\gamma) = -\left(\sum_{i=1}^{n} p_{\mathrm{m},i}\right) / \left(\sum_{i=1}^{n} c_i^{-1}\right). \tag{3.9}$$

In particular, $\boldsymbol{u}^* = \boldsymbol{\pi}^I(\boldsymbol{s}^*)$ *and* $\nabla C_i(u_i^*) = \nabla C_j(u_j^*)\,, \forall i, j \in \mathcal{V}$ *. That is,* $\boldsymbol{u}^*$ *at the equilibrium solves the economic dispatch problem* (3.1)*.*

The proof is given in Appendix B.2. The key steps follow the equality at equilibrium and conditions in Lemma 8.

3.3.4 Asymptotic stability guarantees

The next theorem shows that the unique equilibrium achieved by the Controller Design 1 is locally asymptotically stable.

Theorem 4 (Stability). *Suppose assumptions in Theorem 3 hold. The closed-loop system* (2.3) *is locally asymptotically stable at the unique equilibrium characterized by* (3.8)*.*

We prove that the equilibrium is asymptotically stable by constructing a Lyapunov function $V_2(\boldsymbol{\delta}, \boldsymbol{\omega}, \boldsymbol{s})$ using (2.7) as well as the integral functions

$$R(\boldsymbol{s}) := \sum_{i=1}^{n} \int_0^{s_i} \pi^I(z)\mathrm{d}z \tag{3.10}$$

associated with the monotone function $\pi^I(\cdot)$ and $\psi_l(\cdot)$ in Controller Design 1. Namely, we construct a function

$$V_2(\boldsymbol{\delta}, \boldsymbol{\omega}, \boldsymbol{s}) := V(\boldsymbol{\delta}, \boldsymbol{\omega}) + B(\boldsymbol{s}, \boldsymbol{s}^*)\,, \tag{3.11}$$

where $B(\boldsymbol{s}, \boldsymbol{s}^*)$ is the Bregman distances associated with the integral functions $R(\boldsymbol{s})$, i.e.,

$$B(\boldsymbol{s}, \boldsymbol{s}^*) := R\left(\boldsymbol{s}\right) - R\left(\boldsymbol{s}^*\right) - \nabla R(\boldsymbol{s}^*)^\top \left(\boldsymbol{s} - \boldsymbol{s}^*\right). \tag{3.12}$$

The Bregman distance $B(\boldsymbol{s}, \boldsymbol{s}^*)$ is lower bounded by quadratic forms due to the following lemma.

Lemma 9 (Bregman distances of monotone functions). *For $\pi^I(\cdot) : \mathbb{R} \mapsto \mathbb{R}$ that are Lipschitz continuous and strongly increasing, there exist some $\epsilon_{\mathrm{v}} > 0$ such that the Bregman distances in* (3.12) *satisfy*

$$B(\boldsymbol{s}, \boldsymbol{s}^*) \geq \frac{\epsilon_{\mathrm{v}}}{2} ||\boldsymbol{s} - \boldsymbol{s}^*||_2^2 . \tag{3.13}$$

Proof. We begin by showing that $R(\boldsymbol{s})$ defined in (3.10) is strongly convex. Since $\pi^I(\cdot)$ is strongly increasing, there exists $\epsilon_i > 0$ such that

$$\left(\pi^I(s_i) - \pi^I(s_i')\right)(s_i - s_i') \geq \epsilon_i (s_i - s_i')^2 , \forall s_i, s_i' \in \mathbb{R} . \tag{3.14}$$

Then, note that, $\forall \boldsymbol{s} \neq \boldsymbol{s}'$,

$$\begin{aligned} & (\nabla R(\boldsymbol{s}) - \nabla R(\boldsymbol{s}'))^\top (\boldsymbol{s} - \boldsymbol{s}') \\ = & (\boldsymbol{r}(\boldsymbol{s}) - \boldsymbol{r}(\boldsymbol{s}'))^\top (\boldsymbol{s} - \boldsymbol{s}') \\ = & \sum_{i=1}^{n} \left(\pi^I(s_i) - \pi^I(s_i')\right)(s_i - s_i') \\ \geq & \sum_{i=1}^{n} \epsilon_i (s_i - s_i')^2 \geq \underbrace{\min_{i \in [n]} \epsilon_i}_{:= \epsilon_{\mathrm{v}}} \|\boldsymbol{s} - \boldsymbol{s}'\|_2^2 , \end{aligned} \tag{3.15}$$

where the first inequality results from (3.14). By [64, Chapter IV, Theorem 4.1.4], (3.15) indicates that $R(\boldsymbol{s})$ is ϵ_{v}-strongly convex, which further implies that $B(\boldsymbol{s}, \boldsymbol{s}^*)$ defined in (3.12) satisfies (3.13) by [64, Chapter IV, Theorem 4.1.1].

□

The time derivative of $B(\boldsymbol{s}, \boldsymbol{s}^*)$ is

$$\begin{aligned} & \dot{B}(\boldsymbol{s}, \boldsymbol{s}^*) \\ & = \left(\nabla R(\boldsymbol{s}) - \nabla R(\boldsymbol{s}^*)\right)^\top \dot{\boldsymbol{s}} \\ & \overset{①}{=} \left(\boldsymbol{\pi}^I(\boldsymbol{s}) - \boldsymbol{\pi}^I(\boldsymbol{s}^*)\right)^\top \left((\boldsymbol{\omega} - \boldsymbol{\omega}^*) - \hat{\boldsymbol{c}} \tilde{\boldsymbol{E}} \boldsymbol{\phi} \left(\tilde{\boldsymbol{E}}^\top \nabla \boldsymbol{C}(\boldsymbol{\pi}^I(\boldsymbol{s}))\right)\right), \end{aligned} \tag{3.16}$$

where ① follows from $\nabla R(\boldsymbol{s}) = \boldsymbol{\pi}^I(\boldsymbol{s})$ and $\dot{\boldsymbol{s}} = \left((\boldsymbol{\omega} - \boldsymbol{\omega}^*) - \hat{\boldsymbol{c}} \tilde{\boldsymbol{E}} \boldsymbol{\phi} \left(\tilde{\boldsymbol{E}}^\top \nabla \boldsymbol{C}(\boldsymbol{\pi}^I(\boldsymbol{s}))\right)\right)$ by Controller Design 1.

The next Lemma shows that $\boldsymbol{\pi}^I(\boldsymbol{s}^*)^\top \hat{\boldsymbol{c}} \tilde{\boldsymbol{E}} \boldsymbol{\phi} \left(\tilde{\boldsymbol{E}}^\top \nabla \boldsymbol{C}(\boldsymbol{\pi}^I(\boldsymbol{s}))\right)$ has no impact on the sign of $\dot{V}$.

Lemma 10. *Suppose Assumption 1 holds. Then*

$$\boldsymbol{\pi}^I\left(\boldsymbol{s}^*\right)^\top \hat{\boldsymbol{c}}\tilde{\boldsymbol{E}}\boldsymbol{\phi}\left(\tilde{\boldsymbol{E}}^\top\nabla\boldsymbol{C}(\boldsymbol{\pi}^I(\boldsymbol{s}))\right)=\mathbb{0}_n. \tag{3.17}$$

Proof. Plugging in $\boldsymbol{\pi}^I\left(\boldsymbol{s}^*\right)=\nabla C_{\mathrm{o}}^{-1}(\gamma)\hat{\boldsymbol{c}}^{-1}\mathbb{1}_n$ in (3.8b) gives

$$\begin{aligned}
&\boldsymbol{\pi}^I\left(\boldsymbol{s}^*\right)^\top \hat{\boldsymbol{c}}\tilde{\boldsymbol{E}}\boldsymbol{\phi}\left(\tilde{\boldsymbol{E}}^\top\nabla\boldsymbol{C}(\boldsymbol{\pi}^I(\boldsymbol{s}))\right)\\
=\;&\nabla C_{\mathrm{o}}^{-1}(\gamma)\mathbb{1}_n^\top(\hat{\boldsymbol{c}}^{-1})^\top\hat{\boldsymbol{c}}\tilde{\boldsymbol{E}}\boldsymbol{\phi}\left(\tilde{\boldsymbol{E}}^\top\nabla\boldsymbol{C}(\boldsymbol{\pi}^I(\boldsymbol{s}))\right)\\
=\;&\nabla C_{\mathrm{o}}^{-1}(\gamma)\mathbb{1}_n^\top\tilde{\boldsymbol{E}}\boldsymbol{\phi}\left(\tilde{\boldsymbol{E}}^\top\nabla\boldsymbol{C}(\boldsymbol{\pi}^I(\boldsymbol{s}))\right),
\end{aligned}$$

which equals to $\mathbb{0}_n$ since $\mathbb{1}_n^\top\tilde{\boldsymbol{E}}=\mathbb{0}_n$. □

The full proof of Theorem 4 is given below.

Proof. With Lemma 9, it is straightforward that $V_2(\boldsymbol{\delta},\boldsymbol{\omega},\boldsymbol{s})$ is positive definite and is a well-defined Lyapunov function.

The time derivative of the Lyapunov function in (3.11) is

$$\begin{aligned}
\dot{V}_2(\boldsymbol{\delta},\boldsymbol{\omega},\boldsymbol{s})&=\dot{V}_1(\boldsymbol{\delta},\boldsymbol{\omega})+\dot{B}(\boldsymbol{s},\boldsymbol{s}^*)\\
&\overset{①}{=}-\begin{bmatrix}\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta})-\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*)\\ \boldsymbol{\omega}-\boldsymbol{\omega}^*\end{bmatrix}^T\boldsymbol{Q}(\boldsymbol{\delta})\begin{bmatrix}\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta})-\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*)\\ \boldsymbol{\omega}-\boldsymbol{\omega}^*\end{bmatrix}\\
&\quad-\left[\boldsymbol{\omega}-\boldsymbol{\omega}^*+\epsilon\left(\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta})-\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*)\right)\right]^T\left(\boldsymbol{u}(\boldsymbol{\omega})-\boldsymbol{u}(\boldsymbol{\omega}^*)\right)\\
&\quad+\left(\boldsymbol{\pi}^I(\boldsymbol{s})-\boldsymbol{\pi}^I\left(\boldsymbol{s}^*\right)\right)^\top\left(\left(\boldsymbol{\omega}-\boldsymbol{\omega}^*\right)-\hat{\boldsymbol{c}}\tilde{\boldsymbol{E}}\boldsymbol{\phi}\left(\tilde{\boldsymbol{E}}^\top\nabla\boldsymbol{C}(\boldsymbol{\pi}^I(\boldsymbol{s}))\right)\right)\\
&\overset{②}{\leq}-\rho V(\boldsymbol{\delta},\boldsymbol{\omega})-\boldsymbol{\pi}^I(\boldsymbol{s})^\top\left(\hat{\boldsymbol{c}}\tilde{\boldsymbol{E}}\boldsymbol{\phi}\left(\tilde{\boldsymbol{E}}^\top\nabla\boldsymbol{C}(\boldsymbol{\pi}^I(\boldsymbol{s}))\right)\right)\\
&\overset{③}{\leq}-\rho V(\boldsymbol{\delta},\boldsymbol{\omega})
\end{aligned} \tag{3.18}$$

where the equality ① uses (3.16). The equality ② uses (2.12) and $\boldsymbol{\pi}^I\left(\boldsymbol{s}^*\right)^\top\hat{\boldsymbol{c}}\tilde{\boldsymbol{E}}\boldsymbol{\phi}\left(\tilde{\boldsymbol{E}}^\top\nabla\boldsymbol{C}(\boldsymbol{\pi}^I(\boldsymbol{s}))\right)=\mathbb{0}_n$ in Lemma 10. The inequality ③ uses $\boldsymbol{\pi}^I(\boldsymbol{s})^\top\left(\hat{\boldsymbol{c}}\tilde{\boldsymbol{E}}\boldsymbol{\phi}\left(\tilde{\boldsymbol{E}}^\top\nabla\boldsymbol{C}(\boldsymbol{\pi}^I(\boldsymbol{s}))\right)\right)\geq 0$ in Lemma 8.

Therefore, $\dot{V}_2(\boldsymbol{\delta},\boldsymbol{\omega},\boldsymbol{s})\leq 0$ with equality only holds at the equilibrium. By Lyapunov conditions, the system is locally asymptotically stable around the equilibrium. □

3.4 Experimental Results

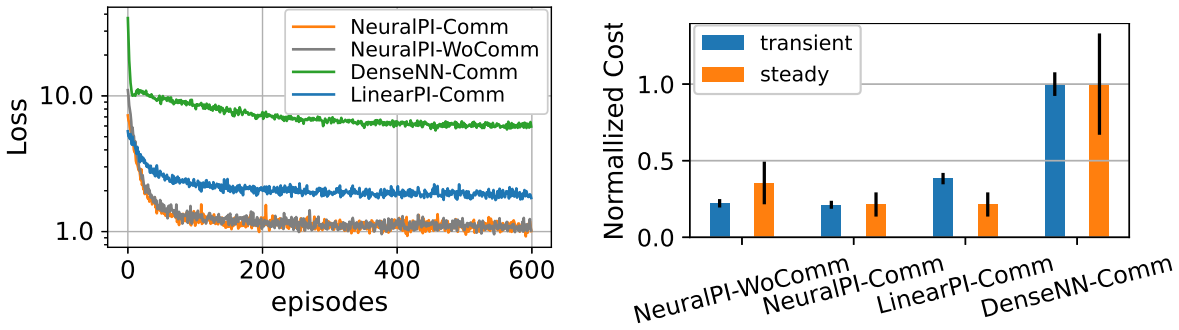
We conduct experiments on the IEEE New England 10-machine 39-bus (NE39) power network with parameters given in [2,65]. We generate the training and test set of size 300 by randomly picking at most three generators to have a step load change uniformly distributed in $\texttt{uniform}[-1, 1]$ p.u., where 1p.u.=100 MW is the base unit of power for the IEEE-NE39 test system. The communication graph is randomly generated to be a regular graph with degree three. The episode number and batch size are 600 and 300, respectively. The step-size in time is set as $\Delta t = 0.01s$ and the number of time stages in a trajectory in the training set is $K = 400$.

3.4.1 Controller performances

We implement frequency control law for power output of generators to realize the agreement of frequency at 60Hz and reduce steady-state power generation cost. Apart from the accumulated frequency deviation, an important metric for the frequency control problem is the maximum frequency deviation (also known as the frequency nadir) after a disturbance. Hence, the transient cost is set to be $J(\boldsymbol{\omega}, \boldsymbol{u}) = \sum_{i=1}^{n} \big(\max_{k=1,\cdots,K} |\omega_i(k\Delta t)| + 0.05 \sum_{k=1}^{K} |\omega_i(k\Delta t)| + \sum_{k=1}^{K} c_i (u_i(k\Delta t))^4 \big)$, where $c_i \sim \texttt{uniform}[0.25, 0.75]$. The steady-state cost in economic dispatch (3.1) is $C(\boldsymbol{u}) = \sum_{i=1}^{n} c_i (u_i^*)^4$, where the cost function is set as the power of four to demonstrate that the proposed approach is not restricted to quadratic cost functions. We use $u_i(30)$ to approximate u_i^* since the dynamics approximately enter the steady state after $t = 30s$ as we will show later in the simulation. The loss function in training is $J(\boldsymbol{\omega}, \boldsymbol{u})$, such that neural networks are optimized to reduce transient cost.

Similar to the case study of the vehicle platoon, we compare the performance of four controllers. The average batch loss during episodes of training is shown in Fig. 3.1(a). All of the four methods converge, with the NeuralPI achieving the lowest cost. Fig. 3.1(b) shows the transient and steady-state costs on the test set. NeuralPI achieves a transient cost that is much lower than the others. Note that the load changes lead to different solutions of the economic dispatch problem, thus the steady-state cost also lies in a range. Still, NeuralPI-Comm and LinearPI-Comm have the

lowest possible steady-state cost, as guaranteed by Theorem 3.

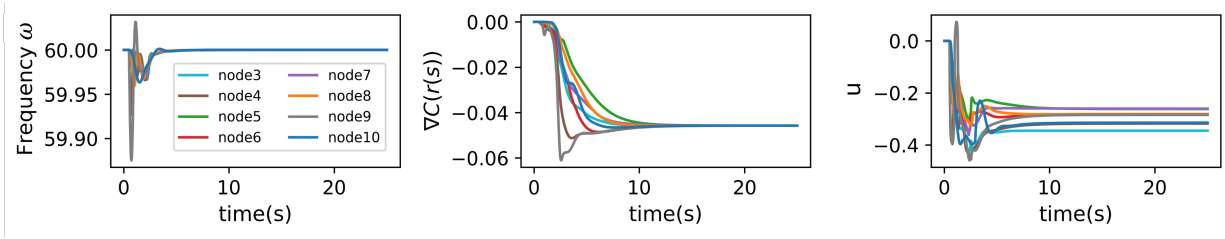


(a) Training Loss

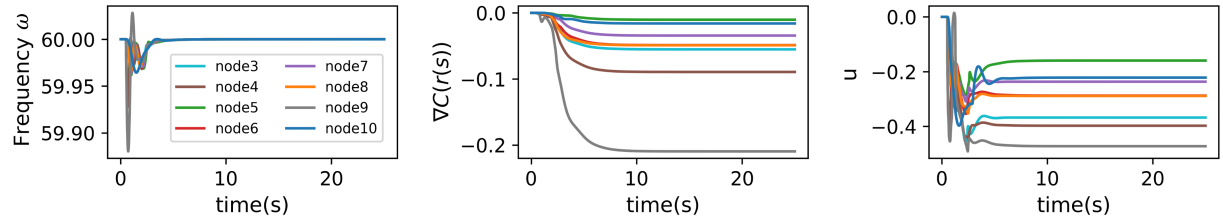
(b) Transient and steady cost

Figure 3.1: (a) Average batch loss along episodes. All converge, with the NeuralPI achieves the lowest cost. (b) The average transient cost and steady-state cost with error bar on the randomly generated test set with size 300. NeuralPI achieves a transient cost that is much lower than others. NeuralPI-Comm and LinearPI-Comm lead to the same lowest steady-state cost guaranteed by Controller Design 1.

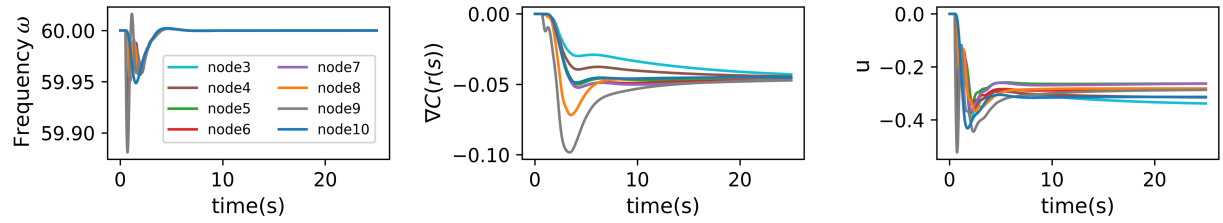
With a step load change at 0.5s, Fig. 3.2 shows the dynamics of frequency $\boldsymbol{\omega}$, marginal cost $\nabla \boldsymbol{C}(\boldsymbol{u})$ and external control action $\boldsymbol{u}$ on 8 nodes under the four methods. As guaranteed by Controller Design 1, NeuralPI-Comm in Fig. 3.2(a) restores the frequency to 60Hz and achieves identical marginal cost, indicating that it achieves the lowest resource allocation cost. NeuralPI-WoComm in Fig. 3.2(b) also reaches the frequency at 60Hz. However, the marginal cost converges at different levels for different nodes because of the lack of communication. LinearPI-Comm in Fig. 3.2(c) converges to the solution with identical marginal cost, but the speed of convergence is slow. DenseNN-Comm in Fig. 3.2(d) exhibits unstable behavior with large oscillations. Hence, the guarantees provided in Controller Design 1 are robust to parameter changes, which have significant practical importance. Controller Design 1 further realizes the economic dispatch of generators under different load levels distributedly.



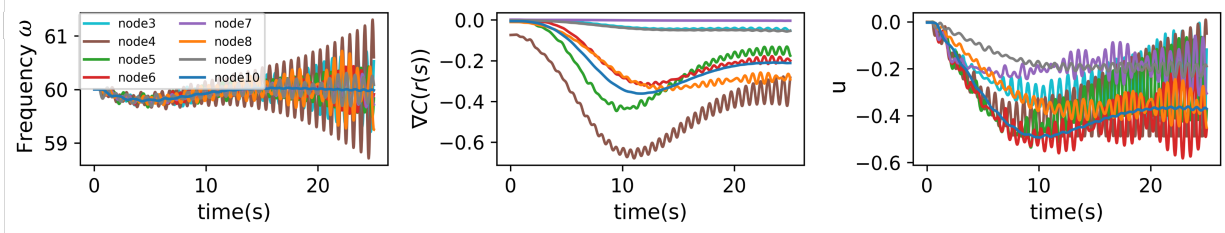
(a) NeuralPI-Comm: dynamics of $\boldsymbol{\omega}$, $\nabla \boldsymbol{C}(\boldsymbol{u})$ and u



(b) NeuralPI-WoComm: dynamics of $\boldsymbol{\omega}$, $\nabla \boldsymbol{C}(\boldsymbol{u})$ and u



(c) LinearPI-Comm: dynamics of $\boldsymbol{\omega}$, $\nabla \boldsymbol{C}(\boldsymbol{u})$ and u



(d) DenseNN-Comm: dynamics of $\boldsymbol{\omega}$, $\nabla \boldsymbol{C}(\boldsymbol{u})$ and u

Figure 3.2: Dynamics of the system under four methods with a step load change at 0.5s. (a) NeuralPI-Comm achieves the output agreement at 60Hz and identical marginal cost. (b) NeuralPI-WoComm achieves the output agreement but fails to converge to the identical-marginal-cost solution. (c) LinearPI-Comm is stable but has slower convergence compared with neural network-based approaches. (d) DenseNN-Comm leads to large frequency deviations and oscillations.

3.5 Conclusion

This chapter proposed structured Neural-PI controllers to achieve provable guarantees on stability and can distributedly achieve optimal resource allocation at the steady state. Experiments demonstrate that the proposed approach can improve both transient and steady-state performances and is also robust to parameter changes, while unstructured neural networks lead to unstable behaviors. The results can also be extended to the control of general networked systems (e.g., vehicle platoons) [66].

Chapter 4

EQUILIBRIUM-INDEPENDENT STABILITY ANALYSIS FOR DISTRIBUTION SYSTEMS WITH LOSSY TRANSMISSION LINES

4.1 Introduction

Distributed energy resources (DERs) such as rooftop solar, electric vehicles and battery storage devices are increasingly entering the power distribution systems. These devices have intermittent outputs and often exhibit large and fast ramping variations, bringing larger disturbances to the system [67, 68]. Therefore, stability of distribution systems under time-varying conditions and large disturbances is becoming a key question in their operations [69].

We are mainly interested in the ability of a system to converge to an acceptable equilibrium following large disturbances [34,70]. In power systems, this is often called transient stability analysis. Most of the time, transmission lines are assumed to be lossless (i.e., the lines are purely inductive with zero resistances), which significantly simplifies the mathematical analysis and allows for explicit constructions of energy functions [2, 24, 71]. However, the transmission lines in distribution systems have non-negligible resistances [72]. More precisely, the r/x ratios of the lines are not very small and the lines are called "lossy" [73, 74]. For lossy systems, transient stability becomes a much harder problem and remains open even for simplified models [69, 75].

A main difficulty in transient stability analysis for lossy networks is the lack of a good Lyapunov function (or energy function) [34, 70]. A classical approach is to use path-dependent integrals to construct Lyapunov functions, but these integrals are not always well-defined and rely on knowing the trajectories of the states [70]. Some works use linear matrix inequalities (LMIs) to find Lyapunov functions by relaxing sinusoidal AC power flow equations [69, 76]. This relaxation can result in very conservative assessments of stability and does not yet scale to moderate or large systems. More recently, attempts have been made to learn a Lyapunov function parameterized by

neural networks [75, 77]. However, it is challenging to verify that the learned neural networks are actually Lyapunov functions.

Apart from the challenges in scalability, existing approaches only apply a single equilibrium at a time [69, 75, 78]. Because of frequent changes to DERs' setpoints, equilibria are time-varying. Hence, it is essential to characterize stability for a set of possible equilibria. In addition, the power electronics on the DERs allow their damping coefficients to be adjusted [16, 77]. But optimizing these coefficients using existing approaches are nontrivial, since they involve solving complicated nonconvex problems. Therefore, the coefficients often are tuned slowly by trial and error, making the design process cumbersome and difficult.

This paper proposes a novel equilibrium-independent approach to transient stability analysis of lossy distribution systems, where we achieve scalability by breaking the network into subsystems. In particular, we consider the angle droop control for the power-electronic interfaces to drive voltage phase angles to their setpoints [69, 75]. For lossy transmission lines, we design a tunable parameter that can serve to explicitly trade off between the control effort and the stability region. At the limit, we recover results for lossless transmission lines, allowing the proposed method to gracefully extrapolates between lossless and lossy systems.

Motivated by equilibrium-independent passivity (EIP) proposed in [79, 80], we study the network stability with time-varying equilibrium points by certifying EIP of each subsystems. Then, stability certification is reduced to checking the diagonal stability property of the interconnection matrix over subsystems subject to EIP conditions. The proposed design of the subsystems divides the interconnection matrix into the summation of a skew-symmetric and a sparse matrix. The stabilizing damping coefficients are then explicitly represented as a convex constraint. This in turn provides a simple yet effective approach to optimize control efforts with guaranteed stability regions. Case studies verify that the proposed method is much less conservative and much more scalable to large systems compared with existing methods [69, 75].

4.2 Model and Problem Formulation

4.2.1 Power-Electronic Interfaced Distribution Systems

Consider a distribution system with n buses and m lines modelled as a connected graph $(\mathcal{N}, \mathcal{L})$, where each bus is equipped with with a power-electronic interface [69, 75] Buses are indexed by $k \in \mathcal{N} := \{1, \ldots, n\}$. Lines are indexed by $l \in \mathcal{L} := \{n+1, \cdots, n+m\}$. Without loss of generality, we define the power flow from i to j to be the positive direction if $i < j$. We denote the interconnections between buses i, j and line l connecting them as $l \in \mathcal{B}_i^+$ and $l \in \mathcal{B}_j^-$, where $\mathcal{B}_i^+$ and $\mathcal{B}_j^-$ represents the line l leaving bus i and entering bus j, respectively.

We adopt the model proposed in [69] where angle and voltage droop control are utilized for real and reactive power sharing through power-electronic interfaces. Let δ_k and v_k be the voltage phase angle and voltage magnitude at bus $k \in \mathcal{N}$, and δ_k^*, v_k^* be their setpoint values set by distribution system operators (for more information on how the setpoints are chosen, see [69, 75]). Let p_k and q_k denote real and reactive power injections at bus k, and p_k^* and q_k^* be their setpoints. The dynamics of bus k are described by

$$\tau_{\mathrm{a}k}\dot{\delta}_k = -d_{ak}(\delta_k - \delta_k^*) + (p_k^* - p_k) \tag{4.1a}$$

$$\tau_{\mathrm{v}k}\dot{v}_k = -d_{vk}(v_k - v_k^*) + (q_k^* - q_k)\,, \tag{4.1b}$$

where $\tau_{\mathrm{a}k}$ and $\tau_{\mathrm{v}k}$ are time constants for voltage phase angle and voltage magnitude at bus k, respectively. The parameters d_{ak} and d_{vk} are damping coefficients controlling power injected by inverters, and thus larger values correspond to larger control efforts. Importantly, the equilibria of the system come from the setpoints δ^* and v_k^*, which are time varying and not known ahead of time.

We follow the model in [69, 75] where $\tau_{vk} \gg \tau_{ak}$ by design. Then, the voltage v_k evolves much slower than the phase angle δ_k, hence the angle and voltage dynamics separates in timescale and v_k is typically assumes to be constant. We therefore focus on the angle stability dynamics in (4.1a) and set $v_k = 1$ per unit in the rest of this paper.

Let g_l and b_l be the conductance and susceptance of the transmission line $l \in \mathcal{L}$, respectively.

The active power flow in the line l from bus i to j is

$$p_l = g_l - g_l \cos(\delta_i - \delta_j) + b_l \sin(\delta_i - \delta_j), \tag{4.2}$$

which is the nonlinear AC power flow equations. We often use δ_{ij} as a shorthand for $\delta_i - \delta_j$. System operators calculate the setpoints such that p_k^* and δ_k^* satisfy the power flow equation for all $k \in \mathcal{N}$. A transmission line is called lossless if $g_l = 0$ and lossy otherwise. For distribution systems, g_l is typically not significantly smaller than b_l.

The buses are interconnected with transmission lines and the active power injected from bus k to the network is

$$p_k = \sum_{l \in \mathcal{B}_k^+} p_l - \sum_{l \in \mathcal{B}_k^-} p_l \quad . \tag{4.3}$$

The dynamics of the system is described by (4.1a), (4.2) and (4.3). The transient stability of the system is defined as the ability to converge to the equilibrium points δ^* from different initial conditions. Since equilibria are set by system operators, the system needs to be stable for multiple possible equilibria. In this paper, we adopt a modular approach to certify stability and design the damping coefficients d_{ak}'s, and show how it overcomes the challenges of existing approaches.

4.2.2 Stability Analysis Through A Modular Approach

The goal of this paper is to answer two key questions for the transient stability of distribution systems: *1) How large is the stability region?* and *2) What is the control effort needed to attain certain range of stability region?* To this end, we certify network-level stability by breaking the network into subsystems. Then by looking at the equilibrium-independent passivity (EIP) of each subsystems and their interconnections, the stability analysis scale to large networked systems with time-varying equilibrium points [80].

For each bus (4.1a) and each transmission line (4.2), we abstract them as a subsystem G_i with input $\boldsymbol{u}_i$ and output $\boldsymbol{y}_i$. Fig. 4.1 shows the diagram for the connection of subsystems. The coupling of the input and output of each subsystems are described by $\boldsymbol{u} = \boldsymbol{M}\boldsymbol{y}$, where the matrix $\boldsymbol{M}$ is determined by interconnections of the system. We show that $\boldsymbol{M}$ is the summation of a skew-

symmetric matrix $\boldsymbol{M}_1$ and a sparse matrix $\boldsymbol{M}_2$. This enable us to obtain a compact and convex expression of stabilizing damping coefficients, which can easily be used for controller design.

Our method gracefully extrapolates between lossless and lossy systems. If all the lines lossless, the sparse component of $\boldsymbol{M}_2$ is zero and only the skew-symmetric part remains. Then standard results from EIP theory can be used to directly show the stability of the system, illustrating why lossless systems are simpler than lossy ones.

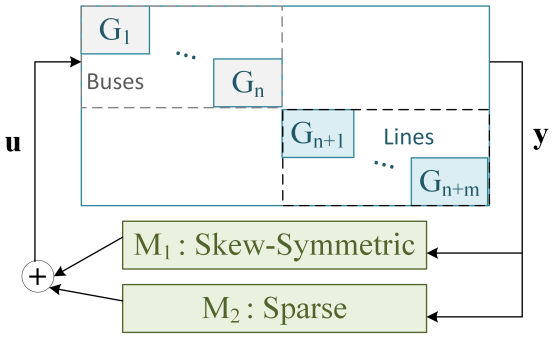


Figure 4.1: Interconnection of buses (grey blocks) and transmission lines (blue blocks). The input and output of each subsystems are interconnected through the $\boldsymbol{y} = (\boldsymbol{M}_1 + \boldsymbol{M}_2)\boldsymbol{u}$, where $\boldsymbol{M}_1$ is skew-symmetric and $\boldsymbol{M}_2$ is sparse.

4.3 Modular Design of Subsystems

With the aim of network stability assessment through the passivity of subsystems, we study the abstraction of (4.1a)-(4.3) as subsystems of buses and lossy transmission lines and their input-output interconnections in this section.

4.3.1 Subsystems for Buses and Lossy Transmission Lines

The subsystem for the lossy transmission line $l \in \mathcal{L}$ leaving bus i and entering bus j is defined with the input $\boldsymbol{u}_l = \begin{bmatrix} \delta_i - \delta_j & \delta_j - \delta_i \end{bmatrix}^\top \in \mathbb{R}^2$ to be the angle differences from i to j and from j to

i. The output $\boldsymbol{y}_l \in \mathbb{R}^2$ is defined to be the modified power flow from i to j and from j to i:

$$\begin{bmatrix} y_{l,1} \\ y_{l,2} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} (g_l - g_l \cos(u_{l,1}))/\alpha_l + b_l \sin(u_{l,1}) \\ (g_l - g_l \cos(u_{l,2}))/\alpha_l + b_l \sin(u_{l,2}) \end{bmatrix} \tag{4.4a}$$

$$\begin{bmatrix} u_{l,1} \\ u_{l,2} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 \\ -1 \end{bmatrix}}_{\Phi_{li}} \delta_i + \underbrace{\begin{bmatrix} -1 \\ 1 \end{bmatrix}}_{\Phi_{lj}} \delta_j \tag{4.4b}$$

where $\alpha_l > 0$ is a tunable scalar and we will study later in detail. At a high level, a larger α_l implies larger stability regions and larger stabilizing damping coefficients. The power flow (4.2) from bus i to j and that from bus j to i can be recovered by $p_{ij} = y_{l,1} - y_{l,2} + \alpha_l(y_{l,1} + y_{l,2})$ and $p_{ji} = -y_{l,1} + y_{l,2} + \alpha_l(y_{l,1} + y_{l,2})$, which will then serve as the input to the subsystem of buses. Stacking the inputs and outputs of lines gives $\boldsymbol{u}_{\mathcal{L}} = [\boldsymbol{u}_{n+1}^\top \cdots \boldsymbol{u}_{n+m}^\top] \in \mathbb{R}^{2m}$, and $\boldsymbol{y}_{\mathcal{L}} = [\boldsymbol{y}_{n+1}^\top \cdots \boldsymbol{y}_{n+m}^\top] \in \mathbb{R}^{2m}$. The matrix block $\Phi_{li} := \begin{bmatrix} 1 & -1 \end{bmatrix}^\top$ and $\Phi_{lj} := \begin{bmatrix} -1 & 1 \end{bmatrix}^\top$ are defined for the mapping from the output of the head i and the tail j to the input of line l, respectively.

The subsystem for bus k is defined with the input $u_k \in \mathbb{R}$ to be the power injection from connected transmission lines and the output $y_k \in \mathbb{R}$ to be the phase angle

$$\tau_k \dot{\delta}_k = -d_k(\delta_k - \delta_k^*) + (P_k^* + u_k) \tag{4.5a}$$

$$y_k = \delta_k \tag{4.5b}$$

$$u_k = \sum_{l \in \mathcal{B}_k^+} \underbrace{\begin{bmatrix} -1 & 1 \end{bmatrix}}_{\Phi_{kl}} y_l + \underbrace{\alpha_l \begin{bmatrix} -1 & -1 \end{bmatrix}}_{\Psi_{kl}} y_l + \sum_{l \in \mathcal{B}_k^-} \underbrace{\begin{bmatrix} 1 & -1 \end{bmatrix}}_{\Phi_{kl}} y_l + \underbrace{\alpha_l \begin{bmatrix} -1 & -1 \end{bmatrix}}_{\Psi_{kl}} y_l \tag{4.5c}$$

where the matrix block Φ_{kl} and Ψ_{kl} is defined for the mapping from the output of the subsystem for line $l \in \mathcal{L}$ to the input of the subsystem for bus $k \in \mathcal{N}$. The matrix block $\Phi_{kl} := \begin{bmatrix} -1 & 1 \end{bmatrix}$ if $l \in \mathcal{B}_k^+$ and $\Phi_{kl} := \begin{bmatrix} 1 & -1 \end{bmatrix}$ if $l \in \mathcal{B}_k^-$. The matrix block $\Phi_{kl} := \begin{bmatrix} -\alpha_k & -\alpha_k \end{bmatrix}$ is defined uniformly for all line l that connects bus k. It will serve to constrain the minimum-effort damping coefficients that stabilize the system.

4.3.2 The Interconnection of Subsystems.

To investigate the stability of the whole interconnected system, we stack the input/output vectors in sequence as $\boldsymbol{u} := (\boldsymbol{u}_{\mathcal{N}}, \boldsymbol{u}_{\mathcal{L}}) \in \mathbb{R}^{n+2m}$ and $\boldsymbol{y} := (\boldsymbol{y}_{\mathcal{N}}, \boldsymbol{y}_{\mathcal{L}}) \in \mathbb{R}^{n+2m}$. The mapping from the output of the bus $k \in \mathcal{N}$ to the input of the line $l \in \mathcal{L}$ is described by a matrix $\boldsymbol{\Phi}_{\mathcal{LN}} \in \mathbb{R}^{2m\times n}$, where the block in the $(2l-1)$-th, $2l$-th row and the k-th column is Φ_{lk} in (4.4). Similarly, the mapping from the output of the line $l \in \mathcal{L}$ to the input of the bus $k \in \mathcal{N}$ is described by the matrix $\boldsymbol{\Phi}_{\mathcal{NL}} \in \mathbb{R}^{n\times 2m}$, where the block in the k-th row and the $(2l-1)$ to $2l$-th column is Φ_{kl} in (4.5). The input-output dependent on α is represented in the matrix $\boldsymbol{\Psi} \in \mathbb{R}^{n\times 2m}$, where the block in the k-th row and the $(2l-1)$ to $2l$-th column is Ψ_{kl} in (4.5). Then, the interconnection of subsystems represented in (4.4) and (4.5) are compactly described by

$$\boldsymbol{u} = (\boldsymbol{M}_1 + \boldsymbol{M}_2)\,\boldsymbol{y} \tag{4.6}$$

where

$$\boldsymbol{M}_1 := \begin{bmatrix} \mathbf{0}_{n\times n} & \boldsymbol{\Phi}_{\mathcal{NL}} \\ \boldsymbol{\Phi}_{\mathcal{LN}} & \mathbf{0}_{2m\times 2m} \end{bmatrix}, \boldsymbol{M}_2 := \begin{bmatrix} \mathbf{0}_{n\times n} & \boldsymbol{\Psi} \\ \mathbf{0}_{2m\times n} & \mathbf{0}_{2m\times 2m} \end{bmatrix}.$$

Note that the matrix $\boldsymbol{\Phi}_{\mathcal{NL}}$ and $\boldsymbol{\Phi}_{\mathcal{LN}}$ is constituted by the blocks that satisfy $\Phi_{il} = -\Phi_{li}^{\top}$ for all $i \in \mathcal{N}$ and $l \in \mathcal{L}$, we have $\boldsymbol{\Phi}_{\mathcal{NL}} + \boldsymbol{\Phi}_{\mathcal{LN}}^{\top} = \mathbf{0}$ and thus $\boldsymbol{M}_1$ is skew-symmetric. The next section will show how the skew-symmetricity of $\boldsymbol{M}_1$ and the sparsity of $\boldsymbol{M}_2$ can be utilized for stability assessment of networked systems.

4.4 Compositional Stability Certification

4.4.1 Equilibrium Independent Passivity.

Equilibrium-independent passivity (EIP), characterized by a dissipation inequality referenced to an arbitrary equilibrium input/output pair, allows one to ascertain passivity of the components without knowledge of the exact equilibrium [79]. The definition is given as follows [79, 80]:

Definition 1 (Equilibrium-Independent Passivity). *The system described by* $\dot{\boldsymbol{\delta}} = f(\boldsymbol{\delta}, \boldsymbol{u}), \boldsymbol{y} = h(\boldsymbol{\delta}, \boldsymbol{u}), \boldsymbol{\delta} \in \mathcal{S}, \boldsymbol{u} \in \mathcal{U}$ *is equilibrium-independent passive if, for every possible equilibrium* $\boldsymbol{\delta}^* \in$

$\mathcal{S}$, there exists a continuously-differentiable storage function $V_{\boldsymbol{\delta}^*} : \mathcal{S} \to \mathbb{R}_{\geq 0}$, such that $V_{\boldsymbol{\delta}^*}(\boldsymbol{\delta}^*) = 0$ and

$$\nabla_{\boldsymbol{\delta}} V_{\boldsymbol{\delta}^*}(\boldsymbol{\delta})^T f(\boldsymbol{\delta}, \boldsymbol{u}) \leq (\boldsymbol{u} - \boldsymbol{u}^*)^\top (\boldsymbol{y} - \boldsymbol{y}^*).$$

If there further exists a positive scalar ϵ such that

$$\nabla_{\boldsymbol{\delta}} V_{\boldsymbol{\delta}^*}(\boldsymbol{\delta})^T f(\boldsymbol{\delta}, \boldsymbol{u}) \leq (\boldsymbol{u} - \boldsymbol{u}^*)^\top (\boldsymbol{y} - \boldsymbol{y}^*) - \epsilon (\boldsymbol{y} - \boldsymbol{y}^*)^\top (\boldsymbol{y} - \boldsymbol{y}^*), \tag{4.7}$$

then the system is strictly EIP.

In the next subsection, we will show that subsystems (4.5) corresponding to the bus $k \in \mathcal{N}$ is strictly EIP in the region $\mathcal{S}_k$ with $\epsilon_k = d_{ak}$ and the storage function $V_k(\boldsymbol{\delta}) = \frac{1}{2\tau_k} \left(\delta_k - \delta_k^*\right)^2$. The subsystem (4.4) corresponding to the line $l \in \mathcal{L}$ is strictly EIP in the region $\mathcal{S}_l$ with $\epsilon_l = \frac{2\alpha_l}{\sqrt{g_l^2 + b_l^2 \alpha_l^2}}$ and the storage function $V_l(\boldsymbol{\delta}) = 0$. We denote $\boldsymbol{\epsilon}_{\mathcal{N}} := (\epsilon_1, \cdots, \epsilon_n)$, $\boldsymbol{\epsilon}_{\mathcal{L}} := (\epsilon_{n+1}\mathbf{1}_2, \cdots, \epsilon_{n+m}\mathbf{1}_2)$ for the EIP coefficients of buses and lines, and the diagonal matrices $\hat{\boldsymbol{\epsilon}}_{\mathcal{L}} := \operatorname{diag}(\boldsymbol{\epsilon}_{\mathcal{L}})$, $\hat{\boldsymbol{\epsilon}}_{\mathcal{N}} := \operatorname{diag}(\boldsymbol{\epsilon}_{\mathcal{N}})$ and $\hat{\boldsymbol{\epsilon}} := \operatorname{diag}(\boldsymbol{\epsilon}_{\mathcal{N}}, \boldsymbol{\epsilon}_{\mathcal{L}}) =$ that will be used in network stability certification. In particular, let $\boldsymbol{d}_{\mathcal{N}} := (d_{a1}, \cdots, d_{an})$, we have $\hat{\boldsymbol{\epsilon}}_{\mathcal{N}} = \operatorname{diag}(\boldsymbol{d}_{\mathcal{N}})$, which links stability certification with the control efforts.

4.4.2 Stability of Interconnected Systems

In this section we derive Lyapunov functions from the storage functions. We define the set $\mathcal{S} := \left\{\bigotimes_{i=1}^{n+m} \mathcal{S}_i\right\}$ to be the states that satisfy strictly EIP for each input-output pairs in all the subsystems. The next lemma allows us to construct Lyapunov functions for any equilibrium that is contained in $\mathcal{S}$. Consequently, $\mathcal{S}$ is a subset of the states where the system will remain stable.

Lemma 11. *Consider the networked system* (4.4)-(4.6) *with input $\boldsymbol{u}$ and output $\boldsymbol{y}$ that interconnected through $\boldsymbol{u} = \boldsymbol{M}\boldsymbol{y}$, where each input-output pair $\{u_i, y_i\}$ is locally strictly EIP with ϵ_i for $\boldsymbol{\delta} \in \mathcal{S}$. If there exists a diagonal matrix $\boldsymbol{C} \succ 0$ such that $\boldsymbol{C}(\boldsymbol{M} - \hat{\boldsymbol{\epsilon}}) + (\boldsymbol{M} - \hat{\boldsymbol{\epsilon}})^\top \boldsymbol{C} \prec 0$, then any equilibrium $\boldsymbol{\delta}^* \in \mathcal{S}$ is locally asymptotically stable.*

Proof. The proof roughly follows [80]. For completeness, we provide the key steps. For the system (4.4)-(4.6), let the sum of the storage functions $V(\boldsymbol{\delta}) = \sum_{i=1}^{n+2m} c_i V_i(\boldsymbol{\delta})$ serve as a candidate

Lyapunov function. Its time derivative is

$$\begin{aligned}
\dot{V}(\boldsymbol{\delta}) &= \sum_{i=1}^{n+2m} c_i \dot{V}_i(\boldsymbol{\delta}) \qquad (4.8)\\
&\leq \sum_{i=1}^{n+2m} c_i \begin{bmatrix} u_i - u_i^* \\ y_i - y_i^* \end{bmatrix}^\top \begin{bmatrix} 0 & 1/2 \\ 1/2 & -\epsilon_i \end{bmatrix} \begin{bmatrix} u_i - u_i^* \\ y_i - y_i^* \end{bmatrix} \\
&= \frac{1}{2} \begin{bmatrix} \boldsymbol{u} - \boldsymbol{u}^* \\ \boldsymbol{y} - \boldsymbol{y}^* \end{bmatrix}^\top \begin{bmatrix} \mathbf{0} & \boldsymbol{C} \\ \boldsymbol{C} & -2\boldsymbol{C}\hat{\boldsymbol{\epsilon}} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} - \boldsymbol{u}^* \\ \boldsymbol{y} - \boldsymbol{y}^* \end{bmatrix} \\
&= \frac{1}{2} \begin{bmatrix} \boldsymbol{y} - \boldsymbol{y}^* \end{bmatrix}^\top \begin{bmatrix} \boldsymbol{M} \\ \boldsymbol{I} \end{bmatrix}^\top \begin{bmatrix} \mathbf{0} & \boldsymbol{C} \\ \boldsymbol{C} & -2\boldsymbol{C}\hat{\boldsymbol{\epsilon}} \end{bmatrix} \begin{bmatrix} \boldsymbol{M} \\ \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{y} - \boldsymbol{y}^* \end{bmatrix} \\
&= \frac{1}{2} (\boldsymbol{y} - \boldsymbol{y}^*)^\top \left(\boldsymbol{C}(\boldsymbol{M} - \hat{\boldsymbol{\epsilon}}) + (\boldsymbol{M} - \hat{\boldsymbol{\epsilon}})^\top \boldsymbol{C} \right) (\boldsymbol{y} - \boldsymbol{y}^*)
\end{aligned}$$

Because $\boldsymbol{y} = \boldsymbol{y}^*$ if and only if $\boldsymbol{\delta} = \boldsymbol{\delta}^*$, $\boldsymbol{C}(\boldsymbol{M} - \hat{\boldsymbol{\epsilon}}) + (\boldsymbol{M} - \hat{\boldsymbol{\epsilon}})^\top \boldsymbol{C} \prec 0$ implies $\dot{V}(\boldsymbol{\delta}) < 0$ for $\boldsymbol{y} \neq \boldsymbol{y}^*$. Hence $V(\boldsymbol{\delta})$ is a valid Lyapunov function for $\boldsymbol{\delta} \in \mathcal{S}$, and an equilibrium $\boldsymbol{\delta}^* \in \mathcal{S}$ is locally asymptotically stable. □

The LMI in Lemma 11 is not jointly convex in $\boldsymbol{d}_{\mathcal{N}}$ or $\boldsymbol{C}$. The next theorem shows how the damping coefficients $\boldsymbol{d}_{\mathcal{N}}$ can be designed based on the special structure of the interconnection matrix $\boldsymbol{M}$.

Theorem 5 (Local Exponential Stability). *If the damping coefficients satisfy* $\operatorname{diag}(\boldsymbol{d}_{\mathcal{N}}) \succ \frac{1}{4}\boldsymbol{\Psi}\hat{\boldsymbol{\epsilon}}_{\mathcal{L}}^{-1}\boldsymbol{\Psi}^\top$, *an equilibrium* $\boldsymbol{\delta}^* \in \mathcal{S}$ *of the system* (4.1)-(4.3) *is locally exponentially stable.*

Proof. This theorem follows from picking $\boldsymbol{C}$ to be the identity matrix. In this case, the condition in Lemma 11 becomes $\left(\boldsymbol{M}^\top + \boldsymbol{M} - 2\hat{\boldsymbol{\epsilon}}\right) \prec 0$. From (4.6), $\boldsymbol{M} = \boldsymbol{M_1} + \boldsymbol{M_2}$, and using the fact that $\boldsymbol{M_1}$ is skew symmetric, and expanding $\hat{\boldsymbol{\epsilon}} := \operatorname{diag}(\boldsymbol{d}_{\mathcal{N}}, \boldsymbol{\epsilon}_{\mathcal{L}})$, we have

$$\dot{V}(\boldsymbol{\delta}) = (\boldsymbol{y} - \boldsymbol{y}^*)^\top \begin{bmatrix} -\operatorname{diag}(\boldsymbol{d}_{\mathcal{N}}) & \frac{1}{2}\boldsymbol{\Psi} \\ \frac{1}{2}\boldsymbol{\Psi} & -\hat{\boldsymbol{\epsilon}_{\mathcal{L}}} \end{bmatrix} (\boldsymbol{y} - \boldsymbol{y}^*). \qquad (4.9)$$

To certify exponential stability, we need to find a scalar $\sigma > 0$, such that $\dot{V}(\boldsymbol{\delta}) < -\sigma V(\boldsymbol{\delta})$. Since the Lyapunov function $V(\boldsymbol{\delta}) = \sum_{i=1}^{n+2m} c_i V_i(\boldsymbol{\delta})$ is

$$\begin{aligned} V(\boldsymbol{\delta}) &= \sum_{i=1}^{n} \frac{1}{2\tau_{ai}} (\delta_i - \delta_i^*)^2 \\ &= (\boldsymbol{y} - \boldsymbol{y}^*)^\top \begin{bmatrix} \frac{1}{2}\operatorname{diag}(\boldsymbol{\tau})^{-1} & \mathbf{0}_{n\times 2m} \\ \mathbf{0}_{2m\times n} & \mathbf{0}_{2m\times 2m} \end{bmatrix} (\boldsymbol{y} - \boldsymbol{y}^*) \end{aligned},$$

then $\dot{V}(\boldsymbol{\delta}) < -\sigma V(\boldsymbol{\delta})$ is equivalent to

$$\begin{bmatrix} 2\operatorname{diag}(\boldsymbol{d}_{\mathcal{N}}) - \sigma \operatorname{diag}(\boldsymbol{\tau})^{-1} & -\boldsymbol{\Psi} \\ -\boldsymbol{\Psi} & 2\hat{\boldsymbol{\epsilon}_{\mathcal{L}}} \end{bmatrix} \succ 0. \tag{4.10}$$

By definition, $\hat{\boldsymbol{\epsilon}_{\mathcal{L}}} \succ 0$ and Schur complement gives

$$\left(2\operatorname{diag}(\boldsymbol{d}_{\mathcal{N}}) - \sigma \operatorname{diag}(\boldsymbol{\tau})^{-1}\right) - \frac{1}{2}\boldsymbol{\Psi}\hat{\boldsymbol{\epsilon}_{\mathcal{L}}}^{-1}\boldsymbol{\Psi}^\top \succ 0.$$

If $\operatorname{diag}(\boldsymbol{d}_{\mathcal{N}}) \succ \frac{1}{4}\boldsymbol{\Psi}\hat{\boldsymbol{\epsilon}}_{\mathcal{L}}^{-1}\boldsymbol{\Psi}^\top$, then any σ satisfying $0 < \sigma < \lambda_{\min}\left(2\operatorname{diag}(\boldsymbol{d}_{\mathcal{N}}) - \frac{1}{2}\boldsymbol{\Psi}\hat{\boldsymbol{\epsilon}}_{\mathcal{L}}^{-1}\boldsymbol{\Psi}^\top\right) \min_{i=1}^{n} \tau_{ai}$ guarantees (4.10) and therefore the equilibrium $\boldsymbol{\delta}^*$ is locally exponentially stable. □

Note that the damping coefficients obtained in Theorem 5 is derived by setting $\boldsymbol{C} = \boldsymbol{I}$, thus the region of stabilizing damping coefficients $\operatorname{diag}(\boldsymbol{d}_{\mathcal{N}}) \succeq \frac{1}{4}\boldsymbol{\Psi}\hat{\boldsymbol{\epsilon}}_{\mathcal{L}}^{-1}\boldsymbol{\Psi}^\top$ is a subset of that verified through $\boldsymbol{C}(\boldsymbol{M} - \hat{\boldsymbol{\epsilon}}) + (\boldsymbol{M} - \hat{\boldsymbol{\epsilon}})^\top \boldsymbol{C} \prec 0$. We will show in the case study that the damping coefficient obtained by $\operatorname{diag}(\boldsymbol{d}_{\mathcal{N}}) \succeq \frac{1}{4}\boldsymbol{\Psi}\hat{\boldsymbol{\epsilon}}_{\mathcal{L}}^{-1}\boldsymbol{\Psi}^\top$ is already much less conservative compared with existing LMIs-based methods [69].

4.5 Controller Design from EIP of Subsystems

In this section, we prove the strictly EIP of the subsystems in (4.4) and (4.5). The system stability region is built from the angles that stabilize each of the subsystems. We also show how each stability region can be tuned to tradeoff with the size of the stabilizing damping coefficients.

4.5.1 Strictly EIP of Lossy Transmission Lines and Buses

The next Lemma shows that the subsystem (4.4) of each lossy transmission line $l \in \mathcal{L}$ is strictly EIP for a region $\mathcal{S}_l$.

Lemma 12 (EIP of Lossy Lines). *The lossy transmission line l from bus i to j represented by* (4.4) *is strictly EIP with $\epsilon_l = \frac{2\alpha_l}{\sqrt{g_l^2+b_l^2\alpha_l^2}}$ for all the possible equilibriums δ_{ij}^* in the set $\mathcal{S}_l = \{\delta_{ij}^* | -\arctan(b_l\alpha_l/g_l) \le \delta_{ij}^* \le \arctan(b_l\alpha_l/g_l)\}$.*

First we note that if $g_l = 0$, then the subsystem (4.4) is strictly EIP in $\delta_{ij}^* \in (-\frac{\pi}{2}, \frac{\pi}{2})$ for any $\alpha_l > 0$. In particular, $\mathbf{\Psi}$ can be made arbitrarily close to $\mathbf{0}$ and $\mathrm{diag}(\boldsymbol{d}_\mathcal{N}) \succ \frac{1}{4}\mathbf{\Psi}\hat{\boldsymbol{\epsilon}}_\mathcal{L}^{-1}\mathbf{\Psi}^\top$ for any $\boldsymbol{d}_\mathcal{N} > 0$. Namely, $\delta_{ij}^* \in (-\frac{\pi}{2}, \frac{\pi}{2})$ is stable for any positive damping coefficients. This recovers the observations for lossless transmission lines [2].

For lossy transmission line with $g_l > 0$, Lemma 12 shows that α_l trades off between the size of $\mathcal{S}_l$ and passivity: a larger α_l enlarges $\mathcal{S}_l$ but also increases the bound $\frac{1}{4}\mathbf{\Psi}\hat{\boldsymbol{\epsilon}}_\mathcal{L}^{-1}\mathbf{\Psi}^\top$ that requires larger damping. The proof is given below.

Proof. The subsystem (4.4) is a memoryless, where $y_{l,1}$ and $y_{l,2}$ is a function of the input $u_{l,1} = \delta_{ij}$ and $u_{l,2} = -\delta_{ij}$, respectively. Hence, it is suffices to consider the function

$$\begin{aligned} y_l(u) =& \frac{g_l - g_l\cos(u)}{2\alpha_l} + \frac{b_l}{2}\sin(u) \\ =& \frac{g_l}{2\alpha_l} + \frac{\sqrt{g_l^2+b_l^2\alpha_l^2}}{2\alpha_l}\sin(u-\gamma_l), \end{aligned} \tag{4.11}$$

when $u = \delta_{ij}$ and $u = -\delta_{ij}$, respectively. The constant $\gamma_l = \arctan(\frac{g_l}{b_l\alpha_l}) \in (0, \pi/2)$ horizontally shift the function $y_l(u)$ as shown in Fig. 4.2 and thus affect the range of δ_{ij} satisfying strictly EIP. For the memoryless system (4.11), we take the storage function to be zero and then the condition for strict passivity is [80]

$$(u-u^*)\left(y_l(u)-y_l(u^*)\right) - \epsilon_l\left(y_l(u)-y_l(u^*)\right)^2 \ge 0, \tag{4.12}$$

which holds for any equilibrium if and only if $y_l'(u) \in \left[0, \frac{1}{\epsilon_l}\right]$. To this end, setting $\epsilon_l = \frac{2\alpha}{\sqrt{g^2+b^2\alpha^2}}$ guarantees that $y_l'(u) \le \frac{1}{\epsilon_l}$. Then $y_l'(u) \ge 0$ is guaranteed for the region $u \in [-\frac{\pi}{2}+\gamma_l, \frac{\pi}{2}+\gamma_l]$, which is labeled in red in Fig. 4.2.

Substituting $u = \delta_{ij}$ and $u = -\delta_{ij}$ gives $-\frac{\pi}{2} \leq \delta_{ij} - \gamma_l \leq \frac{\pi}{2}$ and $-\frac{\pi}{2} \leq -\delta_{ij} - \gamma_l \leq \frac{\pi}{2}$, respectively. Taking the intersection, the angle difference satisfying strictly EIP is $\delta_{ij} \in [-\frac{\pi}{2} + \gamma_l, \frac{\pi}{2} - \gamma_l]$, which is equivalent to $\delta_{ij} \in [-\arctan(b_l\alpha_l/g_l), \arctan(b_l\alpha_l/g_l)]$. □

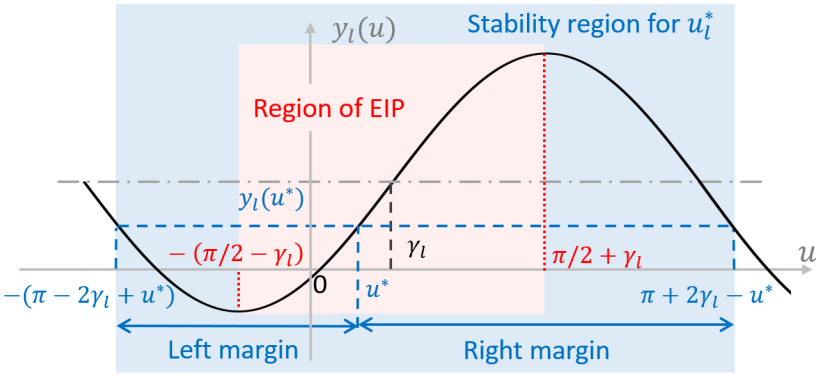


Figure 4.2: The region of EIP $\mathcal{S}_l$ is computed by $y_l'(u) \geq 0$ and is labeled in red. The stability region for an equilibrium u^* is the areas that $y_l(u) - y_l(u^*)$ and $u - u^*$ has the same sign. The stability region is labeled in blue, and its intersection for all equilibrium $u^* \in \mathcal{S}_l$ is the region of EIP in red.

Lemma 13 (EIP of buses). *Bus k represented by* (4.5) *is strictly EIP with $\epsilon_k = d_{ak}$ for all equilibria $\delta_k^* \in \mathbb{R}$.*

This Lemma shows that the subsystem of buses is strictly EIP for all the possible equilibrium of angles. It follows directly from the definitions and we omit the proof.

4.5.2 Sizing Stability Regions

The equilibrium-independent stability guarantees that any equilibrium in the set $\mathcal{S}$ is exponentially stable. Naturally, it is of interest to control the size of the stability region $\mathcal{S}$ (sometimes called region of attraction). The next theorem shows how the parameter α should be chosen if the stability region need to meet a prescribed size.

Theorem 6 (Tuning α for Stability Region). *For the line l from bus i to j with an equilibrium $\delta_{ij}^* \in \mathcal{S}_l$, the stability region is $\mathcal{S}_l|_{\boldsymbol{\delta}^*} = \{\delta_{ij} | -2\arctan(b_l\alpha_l/g_l) - \delta_{ij}^* \le \delta_{ij} \le 2\arctan(b_l\alpha_l/g_l) - \delta_{ij}^*\}$. If $\alpha_l \ge \frac{g_l \tan(|\delta_{ij}^*|+\beta_l/2)}{b_l}$ for a constant $0 < \beta_l < \pi - 2|\delta_{ij}^*|$, then the system is guaranteed to be stable around the equilibrium δ_{ij}^* with at least the margin of β_l, i.e., $[\delta_{ij}^* - \beta_l, \delta_{ij}^* + \beta_l] \subset \mathcal{S}_l|_{\boldsymbol{\delta}^*}$.*

Note that if varying δ_{ij}^* in the set $\mathcal{S}_l = \{\delta_{ij}^* | -\arctan(b_l\alpha_l/g_l) \le \delta_{ij}^* \le \arctan(b_l\alpha_l/g_l)\}$, the intersection of $\mathcal{S}_l|_{\boldsymbol{\delta}^*}$ is exactly $\mathcal{S}_l$. Hence, the region of equilibrium-independent stability can also be understand as the intersection of the stability region for all the possible equilibrium.

Proof. The stability certification (4.8)-(4.10) holds as long as the inequality (4.7) holds. For a certain equilibrium $\boldsymbol{\delta}^*$, we define the stability region $\mathcal{S}_l|_{\boldsymbol{\delta}^*}$ to be the angles satisfying the inequality (4.7). This condition is equivalent to certifying (4.12) for $u = \delta_{ij}$ and $u = -\delta_{ij}$ when fixing $u^* = \delta_{ij}^*$. Note that $\epsilon_l = \frac{2\alpha}{\sqrt{g^2+b^2\alpha^2}}$ gives $y_l'(u) \le \frac{1}{\epsilon_l}$, then condition (4.12) is satisfied as long as $y_l(u) - y_l(u^*)$ is the same sign as $u - u^*$ for both $u = \delta_{ij}$ and $u = -\delta_{ij}$.

The signs of $y_l(u) - y_l(u^*)$ and $u - u^*$ are the same when $u \in [-\pi + 2\gamma_l - u^*, \pi + 2\gamma_l - u^*]$. This region is labeled in blue in Fig. 4.2, which is larger than the region of EIP shown in red. For $u = \delta_{ij}$ and $u = -\delta_{ij}$, we have $\delta_{ij} \in [-\pi + 2\gamma_l - \delta_{ij}^*, \pi + 2\gamma_l - \delta_{ij}^*]$, and $-\delta_{ij} \in [-\pi + 2\gamma_l + \delta_{ij}^*, \pi + 2\gamma_l + \delta_{ij}^*]$, respectively. The intersection gives the region

$$\mathcal{S}_l|_{\boldsymbol{\delta}^*} = \{\delta_{ij} | -\pi + 2\gamma_l - \delta_{ij}^* \le \delta_{ij} \le \pi - 2\gamma_l - \delta_{ij}^*\}. \tag{4.13}$$

and thus $[\delta_{ij}^* - \beta_l, \delta_{ij}^* + \beta_l] \subset \mathcal{S}_l|_{\boldsymbol{\delta}^*}$ yields

$$-\pi + 2\gamma_l - \delta_{ij}^* \le \delta_{ij}^* - \beta_l \le \delta_{ij}^* + \beta_l \le \pi - 2\gamma_l - \delta_{ij}^*,$$

which gives $\frac{\pi}{2} - \gamma_l \ge |\delta_{ij}^*| + \frac{\beta_l}{2}$. Equivalently, $\arctan(\frac{b_l\alpha}{g_l}) \ge |\delta_{ij}^*| + \frac{\beta_l}{2}$ and thus we require $\alpha_l \ge \frac{g_l \tan(|\delta_{ij}^*|+\beta_l/2)}{b_l}$. □

Theorems 5 and 6 provide a way of optimizing over the damping coefficients while guaranteeing the size of the stability region. Specifically, suppose the margin of stable angle difference is $\beta_l \in [0, \pi - 2|\delta_{ij}^*|]$ for $l \in \mathcal{L}$, then we define $\alpha_l = \frac{g_l \tan(|\delta_{ij}^*|+\beta_l/2)}{b_l}$. Thus, $\epsilon_l = \frac{2\alpha_l}{\sqrt{g_l^2+b_l^2\alpha_l^2}}$ and the matrix $\boldsymbol{\Psi}$ is determined by α_l's through (4.5b). To minimize the damping coefficients (corresponding

to hardware costs [81]), we can solve

$$\min_{\boldsymbol{d}_{\mathcal{N}}} \quad \|\boldsymbol{d}_{\mathcal{N}}\|_2 \tag{4.14a}$$

$$\text{s.t. } \operatorname{diag}(\boldsymbol{d}_{\mathcal{N}}) \succ \frac{1}{4}\boldsymbol{\Psi}\hat{\boldsymbol{\epsilon}}_{\mathcal{L}}^{-1}\boldsymbol{\Psi}^{\top}, \tag{4.14b}$$

which is a convex problem. The Pareto-front of the least-cost damping coefficients and the size of stability region can be computed by varying α, quantifying the trade-off between control efforts and stability regions.

4.6 Experimental Results

Case studies are conducted on the IEEE 123-node test feeder [72]. Since existing LMIs-based and neural network-based stability assessment methods all partition the network into a 5-bus system to alleviate computational issues [69, 75], we first work with this 5-bus system as well to show that the proposed method can achieve larger stability regions with smaller damping coefficients. Then, we directly work with the original 123-node feeder to show that the proposed approach can scale to large systems.

4.6.1 Comparison with LMIs-Based Stability Assessment.

We first compare with existing LMI-based transient stability assessment found in [69]. The paramter of the test system (partitioned into 5 buses) can be found in [69, 75]. Under the same damping coefficients, Fig. 4.3 compares the stability region of two lines calculated by our proposed method and the benchmark LMIs-based method in [69]. The angle difference δ_{ij} relative to an equilibrium for the line connecting bus i and j are labeled as $\Delta\delta_{ij} := \delta_{ij} - \delta_{ij}^{*}$. Our proposed approach attains much larger stability region.

From the other direction, if we fix the size of the stability regions, (4.14) can be solved to find the stabilizing damping coefficients. This is in contrast to existing methods, where damping coefficients are found through exhaustive searches.

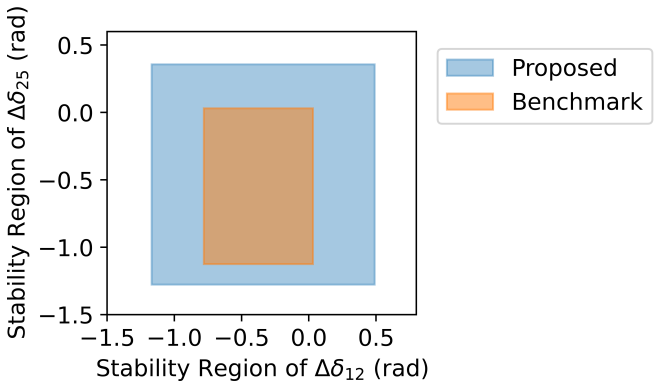
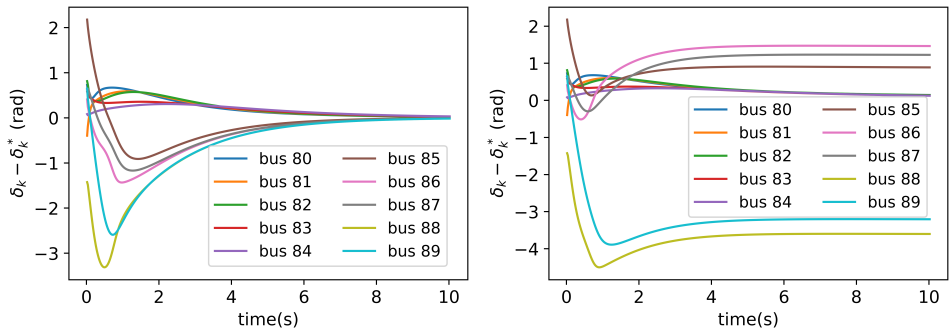


Figure 4.3: Stability regions of $\Delta\delta_{12}$ and $\Delta\delta_{25}$ under the same damping coefficients. The proposed approach finds a larger stability region.

4.6.2 Performance on Large Systems.

To verify the performance of the proposed method on larger systems, we further simulate on the original 123-node test feeder. Fig. 4.4 compares the dynamics of the system with different damping coefficients. The system stabilizes to the setpoints in the former and diverging in the latter case.



(a) $\mathrm{diag}(\boldsymbol{d}_{\mathcal{N}}) \succ \frac{1}{4}\boldsymbol{\Psi}\hat{\boldsymbol{\epsilon}}_{\mathcal{L}}^{-1}\boldsymbol{\Psi}^{\top}$ (b) $\mathrm{diag}(\boldsymbol{d}_{\mathcal{N}}) \prec \frac{1}{4}\boldsymbol{\Psi}\hat{\boldsymbol{\epsilon}}_{\mathcal{L}}^{-1}\boldsymbol{\Psi}^{\top}$

Figure 4.4: Dynamics of ten selected buses with (a) the damping coefficients satisfying the proposed bound in (4.14b) (stable) and (b) the reduced damping coefficients violate the proposed bound (diverges from setpoints).

Moreover, Fig. 4.5 shows the Pareto-front of the width of the stability region and the least-norm

stabilizing damping coefficient by varying α from 0.1 to 2 in the line 1. This quantifies the trade-off between enlarging the stability region and minimizing control efforts.

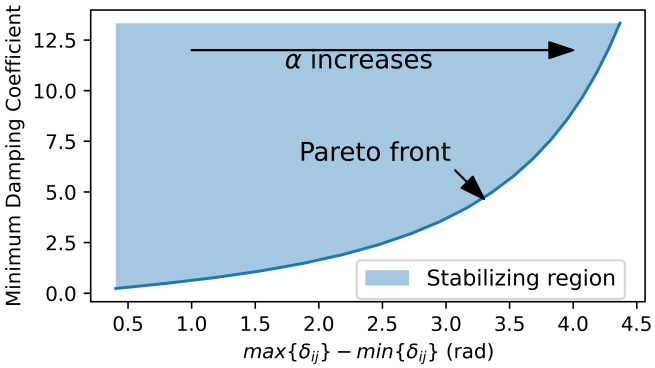


Figure 4.5: Pareto-front of the width of the stability region and the minimum stabilizing damping coefficients by varying α from 0.1 to 2 in line 1.

4.7 Conclusion

This chapter proposes a modular approach for transient stability analysis of distribution systems with lossy transmission lines and time-varying equilibria. Network stability is decomposed into the strictly EIP of subsystems and the diagonal stability of the interconnection matrix. This in turn provides a simple yet effective approach to optimize damping coefficients with guaranteed stability regions. Case studies show that the proposed method is less conservative compared with existing approaches and can scale to large systems. The Pareto-front for the trade-off between stability regions and control efforts can also be efficiently computed.

Part II

DECENTRALIZED AND EFFICIENT LEARNING ALGORITHMS

Chapter 5

DECENTRALIZED SAFE REINFORCEMENT LEARNING FOR INVERTER-BASED VOLTAGE CONTROL

5.1 Introduction

Distributed energy resources (DERs) such as rooftop solar PV, electric vehicles and battery storage are growing at an increasing pace. For example, solar capacity had almost 50% yearly growth in 2021 [82], which is by far the fastest among all renewable resources. Most of these growth are occurring in the distribution network, the low voltage network that connects customers to substations.

High variability of solar PV and sudden change in load due to electric vehicles and storage can lead to large voltage fluctuations. These fluctuations occur at timescales much faster than the conventional mechanical control devices such as tap-changing transformers. Instead, power electronic devices allow flexible and frequent control actions without degrading lifetime. Consequently, there have been growing interests to use the power electronic inverters on the DERs themselves to provide voltage control [83–88].

Since most distribution networks are not yet equipped with real-time communication infrastructure, voltage control strategies should use local measurements available at each bus. More specifically, controllers need to operate at an iterative fashion [89, 90], successively updating their control actions based on each measurement. Designing such decentralized controller is a non-trivial problem. Linear controllers can be far from optimal, even for quadratic costs. Therefore, neural networks have been used to parametrize the controllers to fully utilize the capabilities of the inverters [91–93].

Reinforcement learning algorithms are proposed to train the neural network controllers with trajectory measurements. This provides the advantages of updating neural networks in a model-free

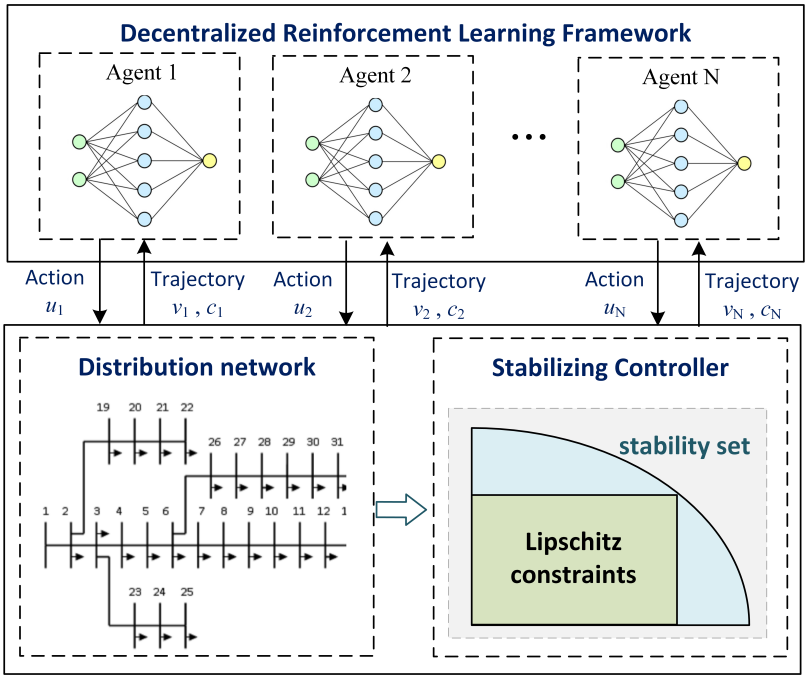


Figure 5.1: Proposed decentralized safe RL approach for optimal voltage control. We prove that the system is guaranteed to be exponentially stable if each controller satisfies certain Lipschitz constraints. The neural network controllers are engineered to satisfy these Lipschitz constraints by design, and is updated from local trajectories with a decentralized RL framework.

setting, i.e., eliminating the requirement on system parameters and communications [22]. Many algorithms, such as deep Q learning [94], actor-critic [95], DDPG [96], have been applied to the control of tap-changing transformers or inverter based resources. Since the control actions are taken in an iterative fashion, it creates a dynamical system, whose transition depends on the actions and the underlying physical distribution network. The key constraint on the controllers is that they do not destabilize the system. However, most works neglect the stability requirement and currently this stability condition is checked through simulations [94, 95]. Considering that voltage control is implemented locally without real-time communication, formal guarantees on stability are required in practice.

This chapter presents a decentralized safe learning method, which guarantees the learned neural network would maintain the stability of iterative voltage control dynamics. We prove that the

system is guaranteed to be exponentially stable if each controller satisfies certain Lipschitz constraints. We optimize the set of Lipschitz bounds to enlarge the search space of controllers. On this basis, we propose to engineer the structure of neural network controllers such that they can satisfy the Lipschitz constraints by design. A decentralized RL framework is constructed to train neural network controller locally at each bus with policy gradient algorithm. The structure of the proposed approach is illustrated in Fig. 5.1.

Case studies show that the controllers learned with stability constraints outperform those with linear controllers and unconstrained neural network controllers. Interestingly, we also observe good learning convergence of the controllers in a model-free setting, even though they interact through the underlying distribution network. Code and data are available at https://github.com/Safe-RL-Power-Systems-Control/Voltage-Control.

5.2 Model

A standard requirement for distribution network is that voltages should deviate no more than 5% from their rated values at all buses [97]. For example, if the rated voltage is 110 V, then the actual voltages should be in the interval from 104.5 V to 115.5 V. For simplicity, we normalize the units such that the reference value for voltage is 1 p.u. For a power network with N buses, let $\boldsymbol{v}$ be the voltage vector where v_i is the voltage at bus i. Let $\boldsymbol{p}$ be active power and $\boldsymbol{q}$ be reactive power. The voltage of the system follows the LinDistFlow model:

$$\boldsymbol{v} = \mathbf{R}\boldsymbol{p} + \mathbf{X}\boldsymbol{q} + \mathbb{1} \tag{5.1}$$

where $\mathbb{1}$ is the all one's vector and $\mathbf{R}$ and $\mathbf{X}$ are positive definite matrices describing the network [89]. The active power depends on external environment and is uncertain and variable. The reactive power comes from phase offsets and is controllable, subject to some actuation constraints [83].

This work focuses on optimizing the control of $\boldsymbol{q}$ through inverter-based resources. The aim of voltage regulation is to control $\boldsymbol{q}$ such that $\boldsymbol{v}$ is close to its reference value. Due to the lack of communication in many distribution systems, $\boldsymbol{q}$ needs to be successively updated based on the

local voltage measurements. Denote $u_i(v_i)$ as the control law for each bus $i = 1, \cdots, N$, which is a mapping from the voltage to reactive power. Let $v_{i,t}$ be the local voltage at the bus i at the t-th iteration step, and denote $\boldsymbol{u}_t = (u_1(v_{1,t}), \cdots, u_N(v_{N,t}))$. We update $\boldsymbol{q}$ and $\boldsymbol{v}$ iteratively as

$$\boldsymbol{q}_{t+1} = \boldsymbol{q}_t - \boldsymbol{u}_t, \tag{5.2a}$$

$$\boldsymbol{v}_{t+1} = \mathbf{R}\boldsymbol{p} + \mathbf{X}\left(\boldsymbol{q}_t - \boldsymbol{u}_t\right) + \mathbb{1}, \tag{5.2b}$$

5.2.1 Optimal voltage control

Our objective is to optimize the $\boldsymbol{u}_t$ to minimize cost in $\boldsymbol{v}$ and $\boldsymbol{q}$ defined as $C(\boldsymbol{u})$, subject to the iterative update rule and the saturation limit on $\boldsymbol{u}_t$. The optimization problem is

$$\min_{\mathbf{u}} \quad C(\boldsymbol{u}) \tag{5.3a}$$

$$\text{s.t. } \boldsymbol{q}_{t+1} = \boldsymbol{q}_t - \boldsymbol{u}_t \tag{5.3b}$$

$$\boldsymbol{v}_{t+1} = \mathbf{R}\boldsymbol{p} + \mathbf{X}\left(\boldsymbol{q}_t - \boldsymbol{u}_t\right) + \mathbb{1} \tag{5.3c}$$

$$\underline{\boldsymbol{u}}_t \leq \boldsymbol{u}_t \leq \overline{\boldsymbol{u}}_t \tag{5.3d}$$

$$\boldsymbol{u}_t \text{ is stabilizing} \tag{5.3e}$$

where constraints (5.3b)-(5.3e) hold for the iteration step t from 0 to T. The cost typically trades off between driving voltage to the reference value and the control effort. The deviation of voltage can typically be quantified as two-norm, one-norm or infinity-norm of the sequence of $\boldsymbol{v}_t$ [85, 98, 99]. The control effort depends on the type of resources and can be both quadratic [13, 14] and non quadratic ones [98–100]. For example, control effort from batteries is commonly defined as one-norm of actions since charging/discharging power affects cycle-depth linearly [99, 100]. The proposed safe RL approach works for all types of cost functions listed above. The lower and upper bound for the control action at bus i are $\underline{u}_{t,i}$ and $\overline{u}_{t,i}$, respectively. The subscript t signifies that these bounds can be time-varying as active power changes.

The controllers $\boldsymbol{u}$ are conventionally designed to be linear (up to a thresholding by (5.3d)), which does not leverage the capability of inverter-based resources in implementing almost arbitrary control laws [16]. To design a flexible non-linear control law for inverter-based resources,

we parameterize each controller $u_i(v_i)$ as a neural network with weight θ_i, sometimes written as $u_{\theta_i}(v_i)$.

However, there remain two challenges. First, due to the lack of communication, neural network controller needs to be trained decentralizely in each bus with local observations of voltages. Second, even if the controller is optimized and implemented locally, they need to be "safe" in the sense that the controller stabilizes the entire system, as defined by (5.3b) and (5.3c). In the next sections, we show how to design the local neural network controllers that guarantee the stability of this system, and how to train the controllers through decentralized reinforcement learning.

In This chapter, we assume that the topology and parameter information of the distribution system is available. That is, we know $\mathbf{X}$, but there is no real-time communication between the buses. This assumption comes from the fact that $\mathbf{X}$ (and $\mathbf{R}$) can be estimated using smart meter data collected over a period of time [101–103], where the communication rate can be quite slow (e.g., once per day [104]). Therefore, design of the controllers can depend on $\mathbf{X}$, but the dependence must be determined offline. The system parameters are not required for real-time training and implementation.

5.3 Stabilizing controller

In this section, we derive the properties of a stabilizing local controller from the Lyapunov stability theory and standard nonlinear system theory. We engineer the structure of neural network to satisfy these structure properties and thus guarantee the stability of the system.

5.3.1 Reduced-order system

We can simplify the dynamics in (5.2) by shifting the origin of the system. Denote $\hat{\boldsymbol{v}}_t$ as the difference between voltage and its reference value $\mathbb{1}$ at time t. We assume that the active power $\boldsymbol{p}$

remains constant during one iteration period. Then we have

$$\begin{aligned}\hat{\boldsymbol{v}}_t &= \boldsymbol{v}_t - \mathbb{1} \\ &= \boldsymbol{R}\boldsymbol{p} + \mathbf{X}\boldsymbol{q}_t \\ &= \boldsymbol{R}\boldsymbol{p} + \mathbf{X}(\boldsymbol{q}_{t-1} - \boldsymbol{u}_{t-1}) \\ &= (\boldsymbol{R}\boldsymbol{p} + \mathbf{X}\boldsymbol{q}_{t-1}) - \mathbf{X}\boldsymbol{u}_{t-1} \\ &= \hat{\boldsymbol{v}}_{t-1} - \mathbf{X}\boldsymbol{u}_{t-1}\end{aligned} \tag{5.4}$$

Therefore, instead of the iteration with both $\boldsymbol{q}_t$ and $\boldsymbol{v}_t$ being the state variables, it suffices to study the dynamics in $\hat{\boldsymbol{v}}_t$.

5.3.2 Structure property of a stabilizing controller

The structure property of a stabilizing controller is obtained from Theorem 7. It shows that as long as each controller u_i satisfies the Lipschitz constraints, the system is guaranteed to be locally exponentially stable.

Theorem 7. *Suppose a vector $\boldsymbol{k} = (k_1, \cdots, k_N)$ satisfies $0 \prec diag(\boldsymbol{k}) \prec 2\mathbf{X}^{-1}$. Then if the derivative of controller satisfies $u_i(0) = 0$ and $0 < \frac{\mathrm{d}u_i(\hat{v}_i)}{\mathrm{d}\hat{v}_i} < k_i$ for all $i = 1, \cdots, N$, the equilibrium point $\boldsymbol{v} = 0$ of the dynamic system in* (5.4) *is locally exponentially stable.*

Proof. The Jacobian of the state transition dynamics in (5.4) is

$$\mathbf{J}(\hat{\boldsymbol{v}}) = \mathbf{I} - \mathbf{X}\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u} \tag{5.5}$$

where $\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u}$ is the gradient of control action $\boldsymbol{u}$ with respect to $\hat{\boldsymbol{v}}$ defined as

$$\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u} = \begin{bmatrix} \frac{\mathrm{d}u_1(\hat{v}_1)}{\mathrm{d}\hat{v}_1} & & \\ & \ddots & \\ & & \frac{\mathrm{d}u_N(\hat{v}_N)}{\mathrm{d}\hat{v}_N} \end{bmatrix}. \tag{5.6}$$

To guarantee an exponentially stable system around the equilibrium, the goal is to show that all the eigenvalues of $\mathbf{J}(\hat{\boldsymbol{v}})$ have magnitude less than 1. To this end, we first show that the eigenvalues of $\mathbf{J}(\hat{\boldsymbol{v}})$ are the same as that of $\mathbf{I} - (\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u})^{\frac{1}{2}}\mathbf{X}(\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u})^{\frac{1}{2}}$.

Let (λ, w) be an eigenpair for $\mathbf{I} - \mathbf{X}(\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u})$. That is, $(\mathbf{I} - \mathbf{X}(\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u}))w = \lambda w$. Then, we have

$$\begin{aligned}
&(\mathbf{I} - (\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u})^{\frac{1}{2}}\mathbf{X}(\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u})^{\frac{1}{2}})(\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u})^{\frac{1}{2}}w \\
=&(\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u})^{\frac{1}{2}}w - (\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u})^{\frac{1}{2}}\mathbf{X}(\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u})w \\
=&(\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u})^{\frac{1}{2}}(\mathbf{I} - \mathbf{X}(\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u}))w \\
=&\lambda(\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u})^{\frac{1}{2}}w
\end{aligned} \tag{5.7}$$

Therefore, $(\lambda, (\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u})^{\frac{1}{2}}w)$ is an eigenpair for $\mathbf{I} - (\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u})^{\frac{1}{2}}\mathbf{X}(\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u})^{\frac{1}{2}}$. To prove that the eigenvalue of $\mathbf{J}(\hat{\boldsymbol{v}})$ to be strictly smaller than 1, it suffices to show that $-\mathbf{I} \prec \mathbf{I} - (\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u})^{\frac{1}{2}}\mathbf{X}(\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u})^{\frac{1}{2}} \prec \mathbf{I}$.

By picking the controller $\boldsymbol{u}$ such that $0 < \frac{\mathrm{d}u_i(\hat{v}_i)}{\mathrm{d}\hat{v}_i} < k_i$ for all $i = 1, \cdots, N$ and $0 \prec \mathrm{diag}(\boldsymbol{k}) \prec 2\mathbf{X}^{-1}$, we have $0 \prec \nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u} \prec 2\mathbf{X}^{-1}$ and thus $(\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u})^{-1} \succ \frac{1}{2}\mathbf{X}$. Since $\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u} \succ 0$ is diagonal, we then have $(\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u})^{\frac{1}{2}}\mathbf{X}(\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u})^{\frac{1}{2}} \prec 2\mathbf{I}$ and thus $-\mathbf{I} \prec \mathbf{I} - (\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u})^{\frac{1}{2}}\mathbf{X}(\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u})^{\frac{1}{2}} \prec \mathbf{I}$. The right side inequality holds because $\mathbf{X} \succ 0$. □

5.3.3 Optimizing search space for neural network controllers

Note that all the feasible stabilizing $\boldsymbol{u}$ are in a convex set described by $\mathcal{S} = \{\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u} | 0 \prec \nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u} \prec 2\mathbf{X}^{-1}\}$. Since there is no communication between buses during training, each $\frac{\mathrm{d}u_i(\hat{v}_i)}{\mathrm{d}\hat{v}_i}$ needs to be bounded by a separate k_i for bus $i = 1, \cdots, N$. Therefore, the search space for neural network controllers is constrained by the selection of $\boldsymbol{k}$. A uniform bound $k_i = \frac{2}{\lambda_{max}(\mathbf{X})}$ can be found in literatures [89], but it might be too conservative since $\mathcal{S}$ may be much larger than the region described by $\mathcal{D} = \left\{\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u} | 0 \prec \nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u} \prec \frac{2}{\lambda_{max}(\mathbf{X})}\mathbb{1}\right\}$.

Here we show an illustration on a three-bus system (with the first bus as the feeder) where $\mathbf{X} = \begin{bmatrix} 0.20 & -0.16 \\ -0.16 & 0.97 \end{bmatrix}$. For different Lipschitz bounds on controllers, feasible regions for $\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u}$ are shown in Fig. 5.2.

The blue area demonstrates the space of controllers constrained by $\mathcal{S} = \{\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u} | 0 \prec \nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u} \prec 2\mathbf{X}^{-1}\}$. The orange area is the space defined by $\mathcal{D} = \left\{\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u} | 0 \prec \nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u} \prec \frac{2}{\lambda_{max}(\mathbf{X})}\mathbb{1}\right\}$, which is the largest square within blue region but is only a very small subset of blue area for $\mathcal{S}$. Note that the axes are scaled so the orange one does not look like a square. With each controller being trained independently, it is natural to consider some larger non-uniform search space such as the green area by

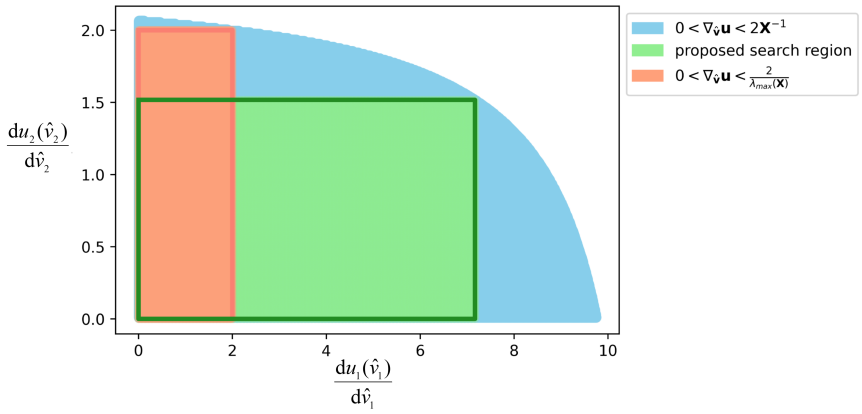


Figure 5.2: Feasible search space comparisons for controllers. The blue area is the set of all feasible $\boldsymbol{u}$ in $\mathcal{S} = \{\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u} | 0 \prec \nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u} \prec 2\mathbf{X}^{-1}\}$. The orange area is the search space with uniform Lipschitz bounds defined as $\mathcal{D} = \left\{\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u} | 0 \prec \nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u} \prec \frac{2}{\lambda_{max}(\mathbf{X})}\mathbf{1}\right\}$, which is the largest square within blue region but is only a very small subset of $\mathcal{S}$. With each controller being trained independently, it is natural to consider some larger non-uniform search space such as the green area.

choosing different $\boldsymbol{k}$. We may choose a $\boldsymbol{k}^*$ such that the search space $\{\nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u} | 0 \prec \nabla_{\hat{\boldsymbol{v}}}\boldsymbol{u} \prec \boldsymbol{k}^*\}$ is the largest rectangular volume inside blue space, denoted as $\prod_{i=1}^{N} k_i$.

The volume is not a convex function in $\boldsymbol{k}$, but we can apply a simple log trick and solve the following optimization problem:

$$\max_{\mathbf{k}} \sum_{i=1}^{N} w_i \log(k_i) \tag{5.8a}$$

$$\text{s.t. } 0 \prec \begin{bmatrix} k_1 & & \\ & \ddots & \\ & & k_N \end{bmatrix} \prec 2\mathbf{X}^{-1} \tag{5.8b}$$

where $w_1, \cdots, w_N$ are the coefficients to represent the relative importance of buses. For example, if bus j has none or very limited capacity for voltage regulation, w_j is set to be small. If bus j is the source node of a branch, w_j can be set to be larger to speed up the convergence of voltage at the source node and thus help the convergence of following branches.

In practice, this set of coefficients can be adjusted according to the solutions of the optimization problem and the training of controllers. For the controller u_i whose derivative $\frac{du_i(\hat{v}_i)}{d\hat{v}_i}$ is far from being bounded by k_i, its coefficient w_i can be adjusted to be smaller to encourage larger control

action at the other buses. We envision that the system operator has the capability to communicate with each bus at a slower timescale (e.g., once a day) and collect the above information. Accordingly, the operator adjusts coefficients w_i, solves (5.8) and issues the bounds k_i to each bus at this the slower timescale.

5.3.4 Design of stabilizing neural network controllers

From Theorem 7, the structural property of locally exponentially stabilizing controllers is derived in Corollary 1. We aim to engineer the neural networks to satisfy these structural property in Corollary 1 by design.

Corollary 1. *The condition for a locally exponentially stabilizing controller in Theorems 7 is equivalent to:*

1. $u_{\theta_i}(\hat{v}_i)$ *has the same sign as* $\hat{v}_i$

2. $u_{\theta_i}(\hat{v}_i)$ *is monotonically increasing*

3. $\frac{\mathrm{d}u_{\theta_i}(\hat{v}_i)}{\mathrm{d}\hat{v}_i} < k_i$.

The first two requirements are equivalent to designing a monotonically increasing function through the origin. This is constructed by decomposing the function into a positive and a negative part as $f_i(\hat{v}_i) = f_i^+(\hat{v}_i) + f_i^-(\hat{v}_i)$, where $f_i^+(\hat{v}_i)$ is monotonically increasing for $\hat{v}_i > 0$ and zero when $\hat{v}_i \leq 0$; $f_i^-(\hat{v}_i)$ is monotonically increasing for $\hat{v}_i < 0$ and zero when $\hat{v}_i \geq 0$. To this end, we formulate the controller with a stacked-ReLU structure shown in Fig. 5.3, which is developed in [2]. This design is a piecewise linear function where the slope of each piece is equal to the summation of weights in activated neurons. Then the requirement 3) can be satisfied by directly

thresholding the slope. The neural network controller is constructed as (5.9)

$$u_i(\hat{v}_i) = \boldsymbol{s}_i \text{ReLU}(\mathbb{1}\hat{v}_i + \boldsymbol{b}_i) + \boldsymbol{z}_i \text{ReLU}(-\mathbb{1}\hat{v}_i + \boldsymbol{d}_i) \tag{5.9a}$$

$$\text{where } 0 < \sum_{j=1}^{l} s_i^j < k_i, \quad \forall l = 1, 2, \cdots, m \tag{5.9b}$$

$$-k_i < \sum_{j=1}^{l} z_i^j < 0, \quad \forall l = 1, 2, \cdots, m \tag{5.9c}$$

$$b_i^1 = 0, b_i^l \leq b_i^{(l-1)}, \quad \forall l = 2, 3, \cdots, m \tag{5.9d}$$

$$d_i^1 = 0, d_i^l \leq d_i^{(l-1)}, \quad \forall l = 2, 3, \cdots, m \tag{5.9e}$$

where m is the number of neurons and $\mathbb{1} \in \mathbb{R}^m$ is the all 1's column vector. Variables $\boldsymbol{s}_i = [s_i^1 \quad s_i^2 \quad \cdots \quad s_i^m]$ and $\boldsymbol{z}_i = [z_i^1 \quad z_i^2 \quad \cdots \quad z_i^m]$ are the weight vector of bus i; $\boldsymbol{b}_i = [b_i^1 \quad b_i^2 \quad \cdots \quad b_i^m]^\intercal$ and $\boldsymbol{d}_i = [d_i^1 \quad d_i^2 \quad \cdots \quad d_i^m]^\intercal$ are the corresponding bias vector. The variables to be trained are weights $\boldsymbol{\theta} = \{\boldsymbol{s}, \boldsymbol{b}, \boldsymbol{z}, \boldsymbol{d}\}$ in (5.9). The saturation limits can be satisfied by hard thresholding the output of the neural network. Note that (5.9) is a single-layer neuron network and m determines the number of pieces for the piece-wise linear function. We tune m according to the testing performances of the controllers and we find that $m = 20$ is generally enough in most settings.

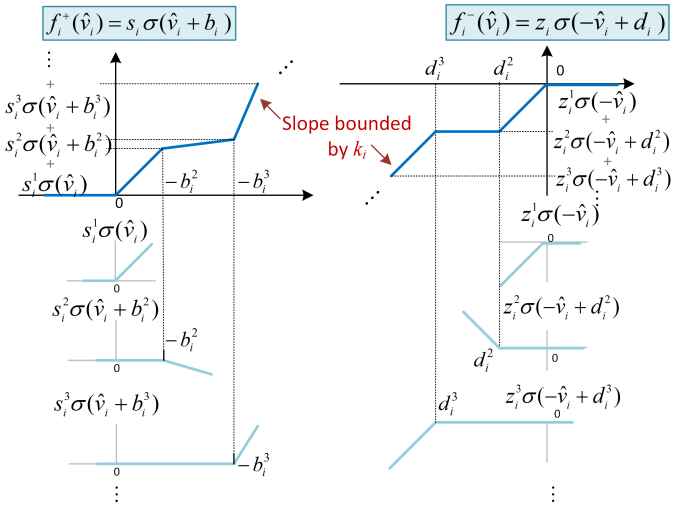


Figure 5.3: Stacked ReLU neural network to formulate a controller satisfying the stabilizing constraint

After one iteration, the Lipschitz constraints guarantee that $||\hat{\boldsymbol{v}}_t|| \leq ||\hat{\boldsymbol{v}}_{t-1}||$ and thus the magnitude of voltage deviation will not be worse after a control action. If there is an abrupt change in active power, the voltage will experience an abrupt change and its magnitude may be larger than before. Therefore, the "safety" in our proposed method is in the sense of stability, where the voltage deviations would go to zero if active power changes relatively slowly. In addition, this guarantees the neural network controller does not degrade the the voltage performance compared to an uncontrolled system.

5.4 Decentralized Safe Reinforcement Learning

In this section, we construct a decentralized reinforcement learning framework to optimize neural network controller in each bus locally with observation of trajectories. By having the constraints in (5.9), the neural network controller is guaranteed to stabilize the system.

Most reinforcement learning algorithms, including Q-learning, actor-critic and DDPG, rely on learning a value function (Q-function) satisfying the Bellman equations. Q-function assumes an infinite-horizon formulation where the states follow a stationary probability distribution, which is generally not true for the voltage control problem in This chapter. Instead, REINFORCE policy gradient algorithm adopts the log probability trick and avoids learning the value function [22]. Therefore, we use REINFORCE policy gradient algorithm to obtain sampled gradient for updating the weights of neural network controllers.

Notably, there are natural noises in the system coming from the changes in active power $\boldsymbol{p}$, which enable us to implement REINFORCE policy gradient with equivalent stochastic policy. Specifically, we assume that the distribution of noise on the system can be estimated. By incorporating noise term into control action, each action $u_{i,t}$ comes from an equivalent stochastic policy with probability distribution $\pi_\theta(u_{i,t}|\hat{v}_{i,t})$. The gradient for updating weights of neural network controller at bus i is obtained by [22]

$$\nabla J(\theta) = \mathbb{E}[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(u_{i,t}|\hat{v}_{i,t}) \sum_{t=1}^{T} C_i(u_{i,t})] \tag{5.10}$$

The pseudo-code for the decentralized RL framework is given in Algorithm 1. Each bus i

has its local RL agent for training in a batch-updating style. Let H be the number of batches. At each episode, each agent collects trajectory $\left\{\hat{v}_{i,1}^h, u_{i,1}^h, \cdots, \hat{v}_{i,T}^h, u_{i,T}^h\right\}$ and the corresponding cost $c_i^h = \sum_{t=1}^T C_i(u_{i,t}^h)$ for $h = 1, \cdots, H$. Adam algorithm is adopted to update weights of neural network controllers with gradient computed through batch average of (5.10). We would also like to emphasize that any model-free RL algorithms can be readily utilized in our framework by replacing the REINFORCE algorithm. We use the standard REINFORCE algorithm in This chapter to illustrate our contributions: the design of stabilizing neural network controllers and training them in a decentralized manner.

Algorithm 2: Decentralized Reinforcement Learning algorithm with Policy Gradient

Require: Learning rate α, batch size H, trajectory length T, number of episodes E
Input: Initial weights θ for control network
for $episode = 1$ *to* E **do**
  for *agent* $i = 1$ *to* N **do**
    Collect trajectories $\left\{\hat{v}_{i,1}^h, u_{i,1}^h, \cdots, \hat{v}_{i,T}^h, u_{i,T}^h\right\}$ and the corresponding cost $c_i^h = \sum_{t=1}^T C_i(u_{i,t}^h)$ for $h = 1, \cdots, H$
    Compute the gradient $\nabla J_i(\boldsymbol{\theta}_i) = \frac{1}{H} \sum_{h=1}^H \sum_{t=1}^T \nabla_\theta \log \pi_\theta(u_{i,t}^h | \hat{v}_{i,t}^h) c_i^h$
    Update weights in the neural network by passing $J_i(\boldsymbol{\theta}_i)$ to Adam optimizer: $\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_i - \alpha \nabla J_i(\boldsymbol{\theta}_i)$
  end
end

5.5 Numerical Results

We verify the performance of the proposed safe RL approach on IEEE 33-bus test feeders [105]. We first show that unconstrained neural network controllers learned by RL might lead to an unstable system, while the controllers trained by safe RL approach are guaranteed to stabilize the system. Then, we show that the proposed decentralized RL framework can learn flexible non-

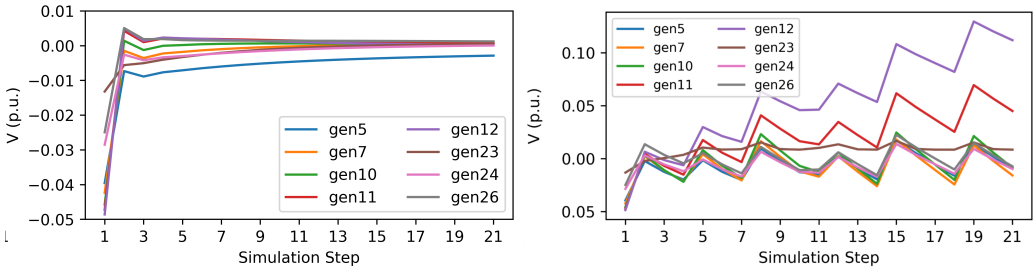
linear controllers for different buses that outperform conventional linear control law.

5.5.1 Simulation setup

The cost function that each controller collectively optimizes is $C(\boldsymbol{u}) = \sum_{t=1}^{T} (||\boldsymbol{v}_t||_1 + \gamma||\boldsymbol{u}_t||_1)$, where γ acts as a trade-off parameter and is set to be 0.01. The base unit for power and voltage is 100kVA and 12.66kV, respectively. The bound on action $\bar{\boldsymbol{u}}$ is generated to be uniformly distributed in $[0.01, 0.05]$. We assume other voltage regulation equipment, such as tap-changing transformers and discrete switching capacitor backs, operate at much slower timescales than the inverters. Therefore, in the simulations, we only consider the operation of the inverters, as learned by an agent running RL algorithms [73, 85]. We use TensorFlow 2.0 framework to build the reinforcement learning environment. The episode number, batch size and the number of neurons are 500, 500, 20, respectively. Parameters of neural network controllers are updated using Adam with learning rate initialized to be 0.003 and decayed every 100 steps with a base of 0.6. We compare the performance of neural network controller designed with and without the safe RL approach, as well as conventional linear controller. All of them are trained using the decentralized RL framework.

5.5.2 Necessity of the stabilizing requirement

Intuitively speaking, if a controller achieves a low loss function after training converges, one might hope that it naturally leads to a stabilizing controller since the trajectory does not blow up to a high cost. Fig. 5.4 shows the dynamics of voltage deviation under the neural network controllers trained with and without the safe RL approach. The one without safe RL approach is *unstable* and leads to very large state oscillations (Fig. 5.4(b)). In contrast, the controller with safe RL approach shows good performance in Fig. 5.4(a). Therefore, explicitly constraining the controller structure is necessary.



(a) safe RL approach

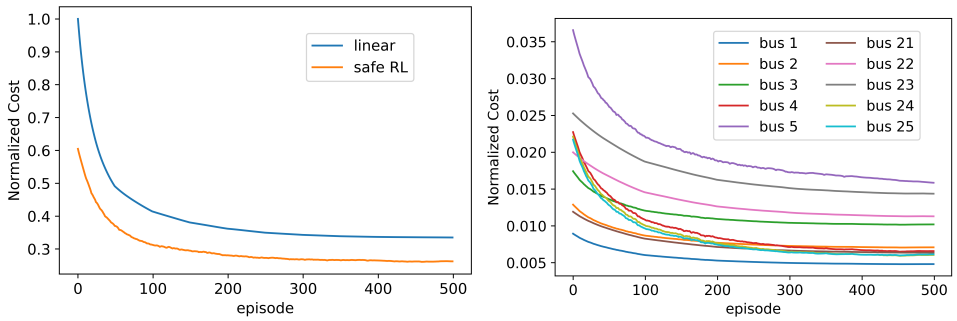
(b) without safe RL approach

Figure 5.4: Dynamics of voltage deviation for safe RL approach(left) and without safe RL approach(right). The controller designed without the safe RL approach leads to unstable trajectories.

5.5.3 Performance comparison

To investigate the convergence of the safe RL approach, Fig. 5.5(a) shows the normalized cost on the test set along episodes for training of neural network controllers and linear controllers. All the losses converge, with the proposed neural network controllers achieving the lowest cost. Fig. 5.5(b) shows the cost on selected buses along the episodes of training. It is interesting to observe that training the controllers in a decentralized fashion did not impact convergence or performance. Namely, during training, u_i is updated based only on the trajectory of v_i, even though the control action impacts the voltage at all neighboring buses.

The control law for neural network controller learned with safe RL, without safe RL approach and linear controller with optimal linear coefficient are shown in Fig. 5.6. The neural network controllers learn flexible non-linear control law for different generators, with the safe RL approach guaranteeing a stabilizing controller by bounding the slope with Lipschitz constraints. Fig. 5.7 illustrates the dynamics of voltage deviation $\boldsymbol{v}$ and corresponding control action $\boldsymbol{u}$ under optimal linear controller and neural network controller trained by safe RL approach. The neural network controller generally leads to faster decay of voltage deviation.



(a) Total cost (b) Cost for selected buses

Figure 5.5: Normalized cost on test set along the episode of training. (a) Total cost during training of neural network controller and linear controller. Neural network controller designed with safe RL approach achieves lower cost than conventional linear controller. (b) Cost during the training of neural network controller. All learning trajectories converge well in the decentralized model-free setting, even though they interact through the underlying distribution network.

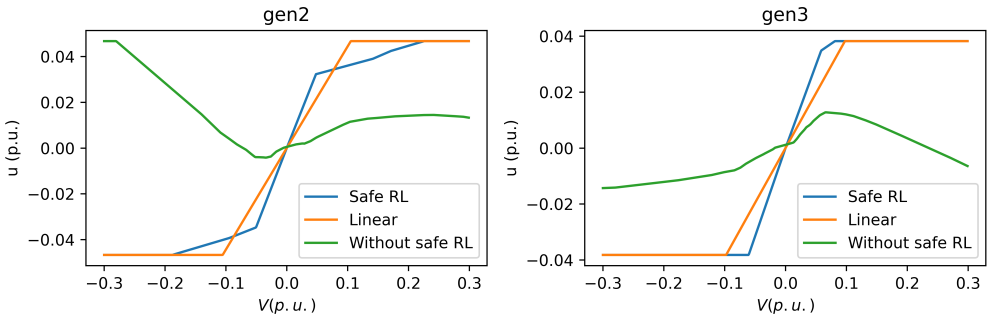
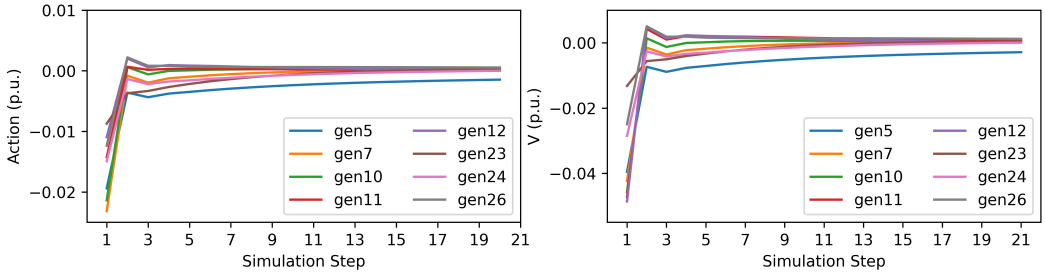
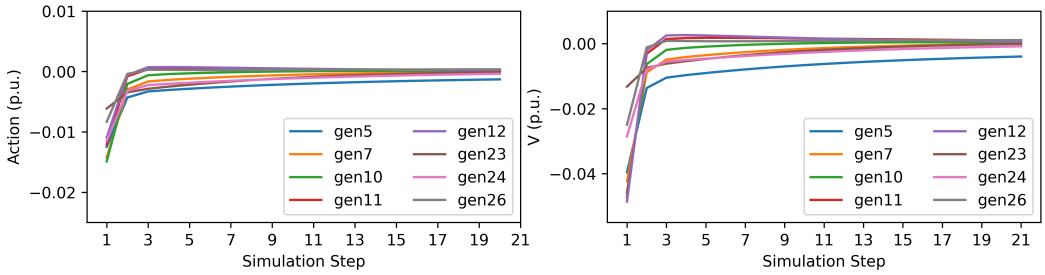


Figure 5.6: Voltage control law obtained by linear controller with optimal linear coefficient, neural network controllers designed with safe RL approach and without safe RL approach. The neural network controllers learn flexible non-linear control laws for different buses, with the slope of controller obtained by safe RL approach bounded by Lipschitz constraints.

In the test set with random initial states, the distribution of cost in selected buses is shown in Fig 5.8. The average costs of the linear controller, the neural network controller bounded by $\frac{2}{\lambda_{max}(\mathbf{X})}$, and the neural network controller with optimal Lipschiz bound obtained in (5.8) are 0.44, 0.38 and 0.36, respectively. Therefore, the proposed approach can learn a stabilizing controller that



(a) Dynamics of $\hat{v}$ (left) and u (right) for neural network controller obtained through safe RL approach



(b) Dynamics of $\hat{v}$ (left) and u (right) for linear control

Figure 5.7: Dynamics of the voltage deviation $\hat{\boldsymbol{v}}$ and the control action $\boldsymbol{u}$ in selected generator buses corresponding to (a) neural network controller trained with safe RL approach (b) Linear control obtained by the same decentralized RL algorithm. The neural network controller generally leads to faster decay of voltage deviation.

reduces the cost by approximately 18.18% compared to conventional linear control law. Moreover, safe RL with the optimal Lipschiz bound also reduces the cost by approximately 5.26% compared to safe RL with the uniform Lipschiz bound $\frac{2}{\lambda_{max}(\mathbf{X})}$.

5.6 Conclusion

This chapter proposes a safe RL approach for optimal voltage control. The exponential stability of the system is guaranteed by controllers constrainted by Lipschitz bounds, which are optimized to enlarge the search space. The neural network controllers are parameterized by a staked ReLU neural network to satisfy stabilizing constraints implicitly. Each bus updates weights locally with

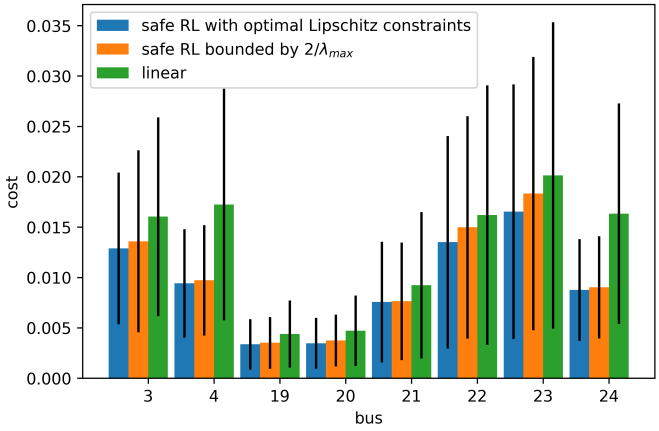


Figure 5.8: Distribution of cost in selected generator buses with random initial states corresponding to safe RL with proposed optimal Lipschiz constraints, safe RL bounded by $\frac{2}{\lambda_{max}(\mathbf{X})}$ and optimal linear control. Compared to uniform bound $\frac{2}{\lambda_{max}(\mathbf{X})}$ and linear controller, the proposed approach reduces the average cost by approximately 5.26%, 18.18%, respectively.

the decentralized RL framework. Case studies show that RL without stability constraints can lead to unstable controllers, while the proposed safe learning approach will lead to a stabilizing controller. The neural network controllers outperform conventional linear controllers by speeding up the convergence of voltages to reference values with relatively low control effort. To further enforce steady-state optimal resource allocation, the Neural-PI structure in Chapter 3 can also be adopted [106].

Chapter 6

EFFICIENT REINFORCEMENT LEARNING THROUGH TRAJECTORY GENERATION

6.1 Introduction

Reinforcement learning (RL) is becoming increasingly popular for the controller design of dynamical systems, especially when the exact system model or parameters are not available [107–109]. Much of the success in RL has relied on sampling-based algorithms such as the policy gradient algorithm [53, 110], which typically requires repeated online interactions with the system. Moreover, the control actions need to incorporate sufficient exploration for the learning algorithm to search for better policies [111]. However, sampling large batches of trajectories is expensive in many real-world problems (e.g., in energy systems, robotics or healthcare), and the exploration requirement for safety-critical systems may be dangerous [112, 113].

When the online interactions with the system are limited, two categories of RL methods are designed: off-policy RL and offline RL. Off-policy RL methods (e.g., Q-learning and its variants) typically learn a quality function (i.e, Q function) leveraging past experience, but online interactions with explorations are still required after the update of the control policies [114]. Offline RL seeks to learn from a fixed dataset without interactions with environments [111, 115]. The fundamental challenge is that once the control policies have been updated, the trajectories of the system under the new policies would not have the same distribution as the historical data [113, 114]. As a result, existing algorithms typically constrain the control policy to be close with the policy utilized in the fixed dataset [113, 116]. Since most algorithms need to do some exploration, it is believed that past data is not helpful if high-reward regions are not covered in the collected trajectories [111, 112].

A fundamental reason behind the above challenges is that the training process is restricted

to fixed trajectories in the historical data, hence RL algorithms need to be restricted to historical control policies. We look at the problem from the other direction: *Using only historical data, can we generate trajectories that follow the same distribution induced by a new control policy?*

This chapter proposes a trajectory generation algorithm for linear systems, which adaptively generates new trajectories as if the system were being operated and explored under the updated control policies. The key insights come from the fundamental lemma for linear systems, which shows that any set of persistently exciting trajectories can be used to represent the input-output behavior of the system [117–119]. Inspired by this, we generate trajectories from linear combinations of historical trajectories, which can come from routine operations of the system. The set of linear combinations is derived from the updated control policy with perturbations on actions, such that the generated trajectory is the same as the trajectory sampled on the real system. This adaptive approach overcomes the challenges in distributional shift and lack of exploration. This is complementary to recent advances in learning linear feedback controllers for linear systems (See, for example, [107, 108, 110, 120] and references within), where trajectories are sampled through online interactions. Experiments show that the proposed method significantly reduces the number of sampled data needed for RL algorithms. We summarize the contributions as follows:

1) We propose a simple end-to-end approach to generate input-output trajectories for linear systems, which significantly reduces the burden of sample collection in RL methods. In Theorem 8, we prove that the generated trajectories is adaptive to the distribution shift of any linear state-feedback controller with perturbations on actions for explorations. When the states are not directly observed, Theorem 9 shows that this framework also applies to output-feedback control by defining an extended state from the observations.
2) The proposed trajectory generation algorithm is compatible with any RL methods that learns from trajectories. The number of samples needed to learn is independent to the batch size (the number of trajectories in each episode) and the number of training episodes.

6.2 Preliminaries and Problem Formulation

6.2.1 Notations

Throughout this manuscript, vectors are denoted in lower case bold and matrices are denoted in upper case bold, unless otherwise specified. Vectors of all ones and zeros are denoted by $\mathbb{1}_n, \mathbb{0}_n \in \mathbb{R}^n$, respectively. The identity matrix is denoted by $\boldsymbol{I}_n \in \mathbb{R}^{n\times n}$. We use $\mathcal{N}(\boldsymbol{A})$ to denote the null space of matrix $\boldsymbol{A}$. Given n matrices $\boldsymbol{M}_i, i = 1, \cdots, n$, we denote $[\boldsymbol{M}_1; \cdots; \boldsymbol{M}_n] := [\boldsymbol{M}_1^\top \cdots \boldsymbol{M}_n^\top]^\top$.

Given a discrete-time signal $\boldsymbol{z}(t) \in \mathbb{R}^d$ for $t = 0, 1, \cdots$, we use $\boldsymbol{z}_{[k,k+T]} \in \mathbb{R}^{Td}$ to denote the vector form of the sequence $\{\boldsymbol{z}(k), \ldots, \boldsymbol{z}(k+T)\}$, and the Hankel matrix $\boldsymbol{Z}_{i,t,N} \in \mathbb{R}^{td\times N}$ as

$$\boldsymbol{Z}_{[k,k+T]} = \begin{bmatrix} \boldsymbol{z}(k) \\ \vdots \\ \boldsymbol{z}(k+T) \end{bmatrix}, \boldsymbol{Z}_{i,t,N} = \begin{bmatrix} \boldsymbol{z}(i) & \boldsymbol{z}(i+1) & \cdots & \boldsymbol{z}(i+N-1) \\ \boldsymbol{z}(i+1) & \boldsymbol{z}(i+2) & \cdots & \boldsymbol{z}(i+N) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{z}(i+t-1) & \boldsymbol{z}(i+t) & \cdots & \boldsymbol{z}(i+t+N-2) \end{bmatrix},$$

where k, i and are integers, and t, N, T are natural numbers. The first subscript of $\boldsymbol{Z}_{i,t,N}$ denotes the time at which the first sample of the signal is taken, the second one the number of samples per each column, and the last one the number of signal samples per each row.

6.2.2 Problem formulation

We consider a discrete-time linear time-invariant system

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) \\ \boldsymbol{y}(k) &= \boldsymbol{C}\boldsymbol{x}(k) \end{aligned} \tag{6.1}$$

with state $\boldsymbol{x} \in \mathbb{R}^n$, action $\boldsymbol{u} \in \mathbb{R}^m$, and output $\boldsymbol{y} \in \mathbb{R}^q$ (sometimes called observations in RL literature). If $\boldsymbol{C}$ is not of full (column) rank, we say that the states are not directly observed. Otherwise, we assume $\boldsymbol{C} = \boldsymbol{I}_n$ and the states are directly observed. We assume that $\boldsymbol{A}$ and $\boldsymbol{B}$ are not known. The matrix $\boldsymbol{C}$ is also unknown if the state is not directly observed. We also make the standard assumption that $(\boldsymbol{A}, \boldsymbol{B})$ is stabilizable and $(\boldsymbol{A}, \boldsymbol{C})$ observable [121].

A trajectory is a sequence of observations and actions of length T, given by $\boldsymbol{\tau} = \{\boldsymbol{y}(0), \boldsymbol{u}(0), \cdots, \boldsymbol{y}(T-1), \boldsymbol{u}(T-1)\}$. The control action $\boldsymbol{u}(k)$ most commonly comes from the control policy conditioned on the observation at time k, written as $\boldsymbol{\pi_\theta}(\boldsymbol{u}(k) \mid \boldsymbol{y}(k))$ with $\boldsymbol{\theta}$ being the parameter for the control policy. Let $c(\boldsymbol{\tau})$ be the cost defined over the trajectory $\boldsymbol{\tau}$. The goal is to optimize the control paramater $\boldsymbol{\theta}$ to minimize the expected cost over trajectories, written as:

$$J(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\tau} \sim p_{\boldsymbol{\pi_\theta}}} c(\boldsymbol{\tau}), \tag{6.2}$$

where $p_{\boldsymbol{\pi_\theta}}$ is the probability distribution of trajectory subject to the policy $\boldsymbol{\pi_\theta}$. The definition of $c(\boldsymbol{\tau})$ varies for different problems. For linear control policy, quadratic costs are most commonly utilized to convert the optimization into classical linear quadratic problems [107, 110]. There are typically not closed-form solutions for (6.2) for other cost functions, e.g., $c(\boldsymbol{\tau}) = \sum_{i=1}^{n} \max_{k=0,\cdots,K-1} |x_i(k)|$ for the frequency control problem in power systems [2]. In this case, gradient-based methods can be utilized to update $\boldsymbol{\theta}$, but the lack of system parameters makes it difficult to compute the gradient $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$.

Direct policy gradient methods. Reinforcement learning (RL) is proposed to update $\boldsymbol{\theta}$ through gradient descent, where the gradient $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ is approximated from trajectories of the system under the control policy $\boldsymbol{\pi_\theta}$. For example, the policy gradient methods in [53] shows that the gradient $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ can be equivalently computed by

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\tau} \sim p_{\boldsymbol{\pi_\theta}}(\boldsymbol{\tau})} \left[c(\boldsymbol{\tau}) \sum_{k=0}^{K-1} \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\pi_\theta}}(\boldsymbol{u}(k) \mid \boldsymbol{y}(k)) \right], \tag{6.3}$$

where K is the length of the trajectory. The terms inside the expectation can be computed purely from observations and actions in the trajectory.

If $\boldsymbol{C} = \boldsymbol{I}_n$ in (6.1), the states are directly measured and the control law is called state-feedback control. Otherwise, the control law is called output-feedback control. Note that if $\boldsymbol{C}$ is not full column rank, then $(\boldsymbol{y}(t), \boldsymbol{u}(t))$ cannot uniquely determine $\boldsymbol{y}(t+1)$. Namely, the system is not a Markov decision process with respect to $(\boldsymbol{y}(t), \boldsymbol{u}(t))$ and it is difficult to use generic RL algorithms based on the quality function $Q((\boldsymbol{y}(t), \boldsymbol{u}(t))$ [122]. For illustration purpose, this chapter focus on the policy gradient algorithm (6.3), and we consider both state and output feedback control.

6.2.3 Approximate gradients with sampled trajectories

When online interactions with the system are limited, computing the expectation in (6.3) is not trivial even when the states are directly observed. In training, the expectation in (6.3) is approximated by the sample average over a large number of trajectories $\boldsymbol{\tau}$ collected from the system under the control policy $\boldsymbol{\pi_\theta}$. Since the distribution $p_{\boldsymbol{\pi_\theta}}(\boldsymbol{\tau})$ depends on the parameter $\boldsymbol{\theta}$, a new set of trajectories need to be collected after each iteration of updating $\boldsymbol{\theta}$. Thus, the number of samples increases with the batchsize (i.e., the number of trajectories in each episode) and the number of training episodes.

We seek to update the control policy using historical trajectories and thus do not need to interact with the system during training. Two challenges arise: *(i) Distribution Shift.* If the control policy changes, the distribution of the historical trajectories would be different from the true distribution $p_{\boldsymbol{\pi_\theta}}(\boldsymbol{\tau})$, potentially resulting in large errors when computing (6.3). *(ii) Exploration.* Most RL methods need to add (sometimes large) perturbations on actions to encourage exploration, but training with a fixed set of historical trajectories may limit exploration.

End-to-End Trajectory Generation. We propose to overcome the challenges of distribution shift and the lack of exploration through generating trajectories from historical data. In this chapter, we focus on learning linear feedback control law $\boldsymbol{u}(k) = \boldsymbol{\theta}\boldsymbol{y}(k)$, with $\boldsymbol{\theta} \in \mathbb{R}^{m\times q}$ being the matrix of trainable parameters. For exploration, the action during training follows the control policy with perturbations $\boldsymbol{w}(k)$ as additive noise, written as

$$\boldsymbol{\pi_\theta}\left(\boldsymbol{u}(k) \mid \boldsymbol{y}(k)\right) := \left\{\boldsymbol{u}(k) = \boldsymbol{\theta}\boldsymbol{y}(k) + \boldsymbol{w}(k), \boldsymbol{w}(k) \sim \mathcal{D}\right\}, \tag{6.4}$$

where $\mathcal{D}$ is the prescribed distribution for the perturbations. The variance of $\mathcal{D}$ is typically initialized to be large and then shrink with the training episode to achieve exploration v.s. exploitation.

We provide a simple end-to-end approach to generate input-output trajectories without system identification, allowing it to extend to systems where the states are not directly observed (i.e., when $\boldsymbol{C}$ is not full column rank). The generated trajectories are guaranteed to have the same distribution as $p_{\boldsymbol{\pi_\theta}}(\boldsymbol{\tau})$ for all $\boldsymbol{\theta}$ and $\mathcal{D}$. We first show the trajectory generation algorithm for state-feedback

control in Section 6.3, then generalize the results to output-feedback control in Section 6.4.

6.3 Trajectory Generation for State-Feedback Control

In this section, we show the conditions when the linear combination of historic trajectories spans all possible trajectories. On this basis, we derive the algorithm to generate trajectories corresponding to updated control policies with perturbations on actions for explorations.

6.3.1 Span of historic trajectories

Let $\boldsymbol{u}_{d,[0,L-1]}$ and $\boldsymbol{x}_{d,[0,L-1]}$ be the a length-L input and state from past trajectory, and let the corresponding Hankel matrix $\mathcal{H} \in \mathbb{R}^{(Tm+Tn)\times(L-T+1)}$ defined as

$$\underbrace{\begin{bmatrix} \boldsymbol{U}_{0,T,L-T+1} \\ \boldsymbol{X}_{0,T,L-T+1} \end{bmatrix}}_{\mathcal{H}} := \left[\begin{array}{cccc} \boldsymbol{u}_d(0) & \boldsymbol{u}_d(1) & \cdots & \boldsymbol{u}_d(L-T) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{u}_d(T-1) & \boldsymbol{u}_d(T) & \cdots & \boldsymbol{u}_d(L-1) \\ \hline \boldsymbol{x}_d(0) & \boldsymbol{x}_d(1) & \cdots & \boldsymbol{x}_d(L-T) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x}_d(T-1) & \boldsymbol{x}_d(T) & \cdots & \boldsymbol{x}_d(L-1) \end{array}\right]. \tag{6.5}$$

By the state-space version of Fundamental Lemma [118] shown below, any linear combination of the columns of the Hankel matrix is a length-T input-state trajectory of (6.1). The proof is provided in [118] and we supplement it in Appendix C.1 for completeness.

Lemma 14 (Fundamental Lemma). *If rank* $\begin{bmatrix} \boldsymbol{U}_{0,T,L-T+1} \\ \boldsymbol{X}_{0,L-T+1} \end{bmatrix} = n + Tm$*, then any length-*$T$ *input/state trajectory of system* (6.1) *can be expressed as* $\begin{bmatrix} \boldsymbol{u}_{[0,T-1]} \\ \boldsymbol{x}_{[0,T-1]} \end{bmatrix} = \begin{bmatrix} \boldsymbol{U}_{0,T,L-T+1} \\ \boldsymbol{X}_{0,T,L-T+1} \end{bmatrix} \boldsymbol{g}$*, where* $\boldsymbol{g} \in \mathbb{R}^{L-T+1}$.

For the rank condition in Lemma 14 to hold, the minimum requirement on the length of the collected trajectory is $L-T+1 = n+Tm$, namely, $L = (m+1)T-1+n$. When the rank condition holds, linear combination of the columns of the Hankel matrix is also a length-T trajectory of the system. Thus, we generate a trajectory of length T using

$$\left[\tilde{\boldsymbol{u}}(0);\cdots;\tilde{\boldsymbol{u}}(T-1);\tilde{\boldsymbol{x}}(0);\cdots;\tilde{\boldsymbol{x}}(T-1)\right]=\underbrace{\begin{bmatrix}\boldsymbol{U}_{0,T,L-T+1}\\ \boldsymbol{X}_{0,T,L-T+1}\end{bmatrix}}_{\mathcal{H}}\boldsymbol{g}. \tag{6.6}$$

For convenience, we adopt the notation $\mathcal{H}_u=\boldsymbol{U}_{0,T,L-T+1}$, $\mathcal{H}_x=\boldsymbol{X}_{0,T,L-T+1}$ in the following sections. To represent the rows of blocks starting from the time $k=0,\cdots,T-1$, we denote

$$\mathcal{H}_x^k:=[\boldsymbol{x}_d(k)\ \boldsymbol{x}_d(k+1)\ \cdots\ \boldsymbol{x}_d(L-T+k)],\ \mathcal{H}_u^k:=[\boldsymbol{u}_d(k)\ \boldsymbol{u}_d(k+1)\ \cdots\ \boldsymbol{u}_d(L-T+k)]. \tag{6.7}$$

6.3.2 Trajectory generation

For generic RL algorithms, a trajectory is sampled from the system that starts from an initial state $\boldsymbol{x}(0)$ and subsequently implements the control policy $\boldsymbol{u}(k)=\boldsymbol{\theta}\boldsymbol{x}(k)+\boldsymbol{w}(k), \boldsymbol{w}(k)\sim\mathcal{D}$ for $k=0,\cdots,T-1$. Given $\boldsymbol{\theta}$, the probability density function of a trajectory is

$$p_{\pi_{\boldsymbol{\theta}}}(\boldsymbol{\tau})=p\left(\boldsymbol{x}(0)\right)\prod_{k=0}^{T-1}p\left(\boldsymbol{\theta}\boldsymbol{x}(k)+\boldsymbol{w}(k)|\boldsymbol{x}(k)\right)p\left(\boldsymbol{x}(k+1)\mid\boldsymbol{x}(k),\boldsymbol{\theta}\boldsymbol{x}(k)+\boldsymbol{w}(k)\right), \tag{6.8}$$

which is uniquely determined by $\boldsymbol{x}(0)$ and the sequence of perturbations $\boldsymbol{w}(0),\boldsymbol{w}(1),\cdots,\boldsymbol{w}(T-1)$.

In the following, we generate trajectories for each updated $\boldsymbol{\theta}$ as if they truly come from the system starting from $\boldsymbol{x}(0)$ under perturbations on actions given by $\boldsymbol{w}(0),\cdots,\boldsymbol{w}(T-1)$. Importantly, we use fixed historic trajectories $\{\boldsymbol{u}_d,\boldsymbol{y}_d\}$ where the Hankel matrix $\mathcal{H}$ in (6.5) satisfies $\text{rank}(\mathcal{H})=n+Tm$, and $\boldsymbol{u}_d$ can come from controllers different from the control policy in (6.4).

The key is to use $\boldsymbol{x}(0)$ and $(\boldsymbol{w}(0),\cdots,\boldsymbol{w}(T-1))$ as extra constraints to find the $\boldsymbol{g}$ in (6.6) that generates the trajectory which follows the same distribution as (6.8). From the control policy $\boldsymbol{u}(k)=\boldsymbol{\theta}\boldsymbol{x}(k)+\boldsymbol{w}(k)$, the trajectory subject to the perturbations $\boldsymbol{w}(0),\boldsymbol{w}(1),\cdots,\boldsymbol{w}(T-1)$ should satisfy

$$\begin{bmatrix}\tilde{\boldsymbol{u}}(0)\\ \vdots\\ \tilde{\boldsymbol{u}}(T-1)\end{bmatrix}=\underbrace{\begin{bmatrix}\boldsymbol{\theta}&&\\ &\ddots&\\ &&\boldsymbol{\theta}\end{bmatrix}}_{\boldsymbol{I}_T\otimes\boldsymbol{\theta}}\begin{bmatrix}\tilde{\boldsymbol{x}}(0)\\ \vdots\\ \tilde{\boldsymbol{x}}(T-1)\end{bmatrix}+\begin{bmatrix}\boldsymbol{w}(0)\\ \vdots\\ \boldsymbol{w}(T-1)\end{bmatrix}. \tag{6.9}$$

Note that the generated initial state is given by $\tilde{\boldsymbol{x}}(0) = \begin{bmatrix} \mathcal{H}_x^0 \end{bmatrix} \boldsymbol{g}$. Combining with (6.9) gives

$$\underbrace{\begin{bmatrix} \mathcal{H}_u - (\boldsymbol{I}_T \otimes \boldsymbol{\theta})\, \mathcal{H}_x \\ \mathcal{H}_x^0 \end{bmatrix}}_{\boldsymbol{G}_{\boldsymbol{\theta}}} \boldsymbol{g} = \begin{bmatrix} \boldsymbol{w}(0) \\ \vdots \\ \boldsymbol{w}(T-1) \\ \boldsymbol{x}(0) \end{bmatrix}. \tag{6.10}$$

Note that the matrix $\boldsymbol{G}_{\boldsymbol{\theta}} \in \mathbb{R}^{(Tm+n)\times(L-T+1)}$ is not a square matrix and its rank is determined by the length of historic trajectory L. When the trajectory is sufficiently long and rank$(\mathcal{H}) = n+Tm$, we have $L - T + 1 > Tm + n$ and thus there might be multiple $\boldsymbol{g}$ where (6.10) holds. We use the minimum-norm solution of (6.10) given by

$$\boldsymbol{g}^* = \boldsymbol{G}_{\boldsymbol{\theta}}^\top \left(\boldsymbol{G}_{\boldsymbol{\theta}} \boldsymbol{G}_{\boldsymbol{\theta}}^\top\right)^{-1} \begin{bmatrix} \boldsymbol{w}(0); \cdots; \boldsymbol{w}(T-1); \boldsymbol{x}(0) \end{bmatrix}. \tag{6.11}$$

In the next Theorem, we prove the existence and uniqueness of the trajectory generated by (6.10). The goal is to show that given $(\boldsymbol{w}(0), \cdots, \boldsymbol{w}(T-1), \boldsymbol{x}(0))$, any $\boldsymbol{g}$ that satisfies (6.10) will generate the same trajectory using $\mathcal{H}\boldsymbol{g}$. So it is suffice to choose the closed-form solution in (6.11).

Theorem 8. *If rank$(\mathcal{H}) = n + Tm$, there exists at least one solution $\boldsymbol{g}$ such that* (6.10) *holds. Given $(\boldsymbol{w}(0), \cdots, \boldsymbol{w}(T-1), \boldsymbol{x}(0))$ and any $\boldsymbol{g}$ that solves* (6.10)*, $\mathcal{H}\boldsymbol{g}^*$ generates the same unique trajectory under the control policy* (6.4) *parameterized by $\boldsymbol{\theta}$.*

Theorem 8 shows that $\mathcal{H}\boldsymbol{g}^*$ generates the unique trajectory that starts from an initial state $\boldsymbol{x}(0)$ and subsequently implements the control policy $\boldsymbol{u}(k) = \boldsymbol{\theta}\boldsymbol{x}(k) + \boldsymbol{w}(k), \boldsymbol{w}(k) \sim \mathcal{D}$. Hence, if we fix $\boldsymbol{\theta}$ in (6.11) and sample $(\boldsymbol{x}(0), \boldsymbol{w}(0), \boldsymbol{w}(1), \cdots, \boldsymbol{w}(T-1))$, then $\mathcal{H}\boldsymbol{g}^*$ will generate a batch of trajectories following the distribution (6.8) corresponding to the parameter $\boldsymbol{\theta}$. After the update of $\boldsymbol{\theta}$ in each episode of training, we generate a new batch of trajectories by updating $\boldsymbol{G}_{\boldsymbol{\theta}}$ in (6.10) and sampling new $(\boldsymbol{x}(0), \boldsymbol{w}(0), \boldsymbol{w}(1), \cdots, \boldsymbol{w}(T-1))$. Thus, the generated trajectories adaptively follow the shifted distribution after updating $\boldsymbol{\theta}$. Importantly, it overcomes the negative perception in the field that there is no possibility to improve explorations beyond past trajectories [112]. By sampling the noises $\boldsymbol{w}(k) \sim \mathcal{D}$, explorations can also be achieved through the generated trajec-

tories. Hence, new trajectories are adaptively generated as if the system were being operated and explored under the updated control policies.

To prove Theorem 8, we first show that the null space of $\boldsymbol{G_\theta}$ is exactly the same as that of the Hankel matrix $\mathcal{H}$. Then, rank$(\mathcal{H}) = n + Tm$ yields rank$(\boldsymbol{G_\theta}) = n + Tm$. The full row rank of $\boldsymbol{G_\theta}$ implies that there exist at least one $\boldsymbol{g}$ where (6.10) holds. The uniqueness of trajectory generated by $\boldsymbol{g}$ follows from the fact that $\boldsymbol{G_\theta}$ and $\mathcal{H}$ have the same null space. Details of the proof is given below.

Proof. We first prove that the null space $\mathcal{N}(\boldsymbol{G_\theta})$ is the same as $\mathcal{N}(\mathcal{H})$ from *(i)* and *(ii)* :

(i) For all $\boldsymbol{q} \in \mathcal{N}(\mathcal{H})$, we have $[\mathcal{H}_x]\boldsymbol{q} = \mathbb{0}_{Tn}$ and $[\mathcal{H}_u]\boldsymbol{q} = \mathbb{0}_{Tm}$. Plugging in $\boldsymbol{G_\theta}$ in (6.10) yields $\boldsymbol{G_\theta q} = \mathbb{0}_{Tm+n}$. Namely, $\boldsymbol{q} \in \mathcal{N}(\boldsymbol{G_\theta})$.

(ii) For all $\boldsymbol{v} \in \mathcal{N}(\boldsymbol{G_\theta})$, $\boldsymbol{G_\theta v} = \mathbb{0}_{Tm+n}$ yields $\mathcal{H}_x^0\boldsymbol{v} = \mathbb{0}_n$ and $\mathcal{H}_u^k\boldsymbol{v} = \hat{\boldsymbol{\theta}}\mathcal{H}_x^k\boldsymbol{v}$ for $k = 0, \cdots, T-1$. Thus, $\mathcal{H}_u^0\boldsymbol{v} = \hat{\boldsymbol{\theta}}\mathcal{H}_x^0\boldsymbol{v} = \mathbb{0}_m$. From $\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k)$, we have $\mathcal{H}_x^{k+1} = A\mathcal{H}_x^k + B\mathcal{H}_u^k$. From $\mathcal{H}_x^0 = \mathbb{0}_n$ and $\mathcal{H}_u^0 = \mathbb{0}_m$, we apply $\mathcal{H}_x^{k+1} = A\mathcal{H}_x^k + B\mathcal{H}_u^k$ and $\mathcal{H}_u^k\boldsymbol{v} = \hat{\boldsymbol{\theta}}\mathcal{H}_x^k\boldsymbol{v}$ alternately. This induces $\mathcal{H}_x^k\boldsymbol{v} = \mathbb{0}_n$ and $\mathcal{H}_u^k\boldsymbol{v} = \mathbb{0}_m$ for $k = 0, \cdots, T-1$. Hence, $\mathcal{H}\boldsymbol{v} = \mathbb{0}_{Tm+Tn}$.

Next, we prove the existence of the solution in (6.10). Note that $\mathcal{H} \in \mathbb{R}^{(Tm+Tn)\times(L-T+1)}$. If rank$(\mathcal{H}) = n + Tm$, then rank$(\mathcal{N}(\mathcal{H})) = (L - T + 1) - (n + Tm)$. Since $\mathcal{N}(\boldsymbol{G_\theta})$ is the same as $\mathcal{N}(\mathcal{H})$, then rank$(\mathcal{N}(\boldsymbol{G_\theta})) = (L - T + 1) - (n + Tm)$. It follows directly that rank$(\boldsymbol{G_\theta}) = (L - T + 1) -$ rank$(\mathcal{N}(\boldsymbol{G_\theta})) = n + Tm$. Note that the number of rows of $\boldsymbol{G_\theta}$ is $n + Tm$, then the full row-rank of rank$(\boldsymbol{G_\theta})$ shows that there exists at least one solution such that (6.10) holds.

Lastly, we show the uniqueness of the generated trajectory. Suppose there exists $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$, which are both solutions of (6.10) and $\mathcal{H}\boldsymbol{g}_1 \neq \mathcal{H}\boldsymbol{g}_2$. Since $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ are both solution of (6.10), then $\boldsymbol{G_\theta g}_1 = \boldsymbol{G_\theta g}_1$ and thus $(\boldsymbol{g}_1 - \boldsymbol{g}_2) \in \mathcal{N}(\boldsymbol{G_\theta})$. On the other hand, $\mathcal{H}\boldsymbol{g}_1 \neq \mathcal{H}\boldsymbol{g}_2$ yields $\mathcal{H}(\boldsymbol{g}_1 - \boldsymbol{g}_2) \neq 0$ and thus $(\boldsymbol{g}_1 - \boldsymbol{g}_2) \notin \mathcal{N}(\mathcal{H})$. This contradicts that $\mathcal{N}(\boldsymbol{G_\theta})$ is the same as $\mathcal{N}(\mathcal{H})$. Hence, $\mathcal{H}\boldsymbol{g}_1 = \mathcal{H}\boldsymbol{g}_2$, namely, the generated trajectories are identical. □

6.3.3 Algorithm

By Theorem 8, using historical data, given $\boldsymbol{x}(0)$ and perturbations on actions $\boldsymbol{w}(0), \cdots, \boldsymbol{w}(T-1)$, a trajectory can be generated as if it comes from sampling the system with the current control policy. The details of the algorithm is illustrated in Algorithm 3. We also use REINFORCE policy gradient [53] in Appendix C.2 as an example to show how to use the trajectory generation algorithm in RL methods. The key benefit of Algorithm 3 is that it is adaptive to the updates of parameter $\boldsymbol{\theta}$ and $\boldsymbol{w}(t)$ for exploration. In particular, $\boldsymbol{w}(t)$ can be sampled from any distribution $\mathcal{D}$, making it versatile for different applications.

Algorithm 3: Trajectory generation for state-feedback control

Data collection: Collect historic measurement of the system and stack each T-length input-output trajectory as Hankel matrix $\mathcal{H}$ shown in (6.5) until $\text{rank}(\mathcal{H}) = n + Tm$

Data generation:*Input* :Hankel matrix $\mathcal{H}$, weights $\boldsymbol{\theta}$ and the distribution $\mathcal{D}$ for the control policy, the batchsize Q (number of the generated trajectories), the distribution $\mathcal{S}_x$ of the initial states[1]

Function `TrajectoryGen(`$\mathcal{H}, \boldsymbol{\theta}, \mathcal{D}, Q, \mathcal{S}_x$`)`**:**

  Plug in $\boldsymbol{\theta}$ to compute $\boldsymbol{G}_{\boldsymbol{\theta}} = \left[\mathcal{H}_u - (\boldsymbol{I}_T \otimes \boldsymbol{\theta})\, \mathcal{H}_x; \mathcal{H}_x^0\right]$

  for $i = 1$ *to* Q **do**

    Sample $\boldsymbol{x}_i(0)$ from $\mathcal{S}_x$. Sample $\{\boldsymbol{w}_i(0), \cdots, \boldsymbol{w}_i(T-1)\}$ from $\mathcal{D}$.

    Compute the coefficient $\boldsymbol{g}_i^* = \boldsymbol{G}_{\boldsymbol{\theta}}^\top \left(\boldsymbol{G}_{\boldsymbol{\theta}} \boldsymbol{G}_{\boldsymbol{\theta}}^\top\right)^{-1} [\boldsymbol{w}_i(0); \cdots; \boldsymbol{w}_i(T-1); \boldsymbol{x}_i(0)]$.

    Generate the i-th trajectory

    $\boldsymbol{\tau}_i := [\tilde{\boldsymbol{u}}_i(0); \cdots; \tilde{\boldsymbol{u}}_i(T-1); \tilde{\boldsymbol{x}}_i(0); \cdots; \tilde{\boldsymbol{x}}_i(T-1);] = \mathcal{H} \boldsymbol{g}_i^*$.

  end

  return $[\boldsymbol{\tau}_1, \cdots, \boldsymbol{\tau}_Q]$

[1]The set of historic initial states in past trajectories can be used to estimate $\mathcal{S}_x$.

6.4 Trajectory Generation for Output-Feedback Control

In this section, we show how to obtain a Markov decision process by defining an extended state using input-output trajectory. The key difference to Section 6.3 is that $\boldsymbol{G_\theta}$ may not be full row rank even when the rank condition on the Hankel matrix holds. Thus, the coefficients for generated trajectories and associated proofs are more nuanced.

6.4.1 Extended states for constructing Markov decision process

Let $\mathcal{O}_{[0,\ell]}(\boldsymbol{A}, \boldsymbol{C}) := \operatorname{col}\left(\boldsymbol{C}, \boldsymbol{C}\boldsymbol{A}, \ldots, \boldsymbol{C}\boldsymbol{A}^{\ell-1}\right)$ be the extended observability matrix. The lag of the system (6.1) is defined by the smallest integer $\ell \in \mathbb{Z}_{\geq 0}$ such that the observability matrix $\mathcal{O}_{[0,\ell]}(\boldsymbol{A}, \boldsymbol{C})$ has rank n, i.e., the state can be reconstructed from ℓ measurements [123].

Let $T_0 \geq \ell$ be the length of a trajectory. Define the extended states as

$$\mathcal{X}(k-1) := \left[\ \boldsymbol{y}(k-T_0); \cdots; \boldsymbol{y}(k-1); \boldsymbol{u}(k-T_0); \cdots; \boldsymbol{u}(k-2)\ \right]. \tag{6.12}$$

Then extending the system transition from time step 0 to T_0 gives

$$\mathcal{X}(k) = \tilde{\boldsymbol{A}}\mathcal{X}(k-1) + \tilde{\boldsymbol{B}}\boldsymbol{u}(k-1) \text{ for } k \geq T_0, k \in \mathbb{Z}, \tag{6.13}$$

which is a Markov decision process in terms of the extended states. Detailed proof and the definition of system transition matrix $(\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{B}})$ is given in Appendix C.3. For the output-feedback control law in (6.4), we have $p\left(\boldsymbol{u}(k)|\mathcal{X}(k)\right) = p\left(\boldsymbol{u}(k)|\boldsymbol{y}(k)\right)$ and it is straightforward to show that policy gradient algorithm using (6.3) still works. The proof is given in Appendix C.4.

By defining the the Hankel matrix as $\mathcal{H} := \begin{bmatrix} \boldsymbol{U}_{0,T,L-T+1} \\ \boldsymbol{Y}_{0,T,L-T+1} \end{bmatrix} \in \mathbb{R}^{(Tm+Tq)\times(L-T+1)}$, the following fundamental Lemma in terms of input-output trajectories holds.

Lemma 15 (Fundamental Lemma [117,119]). *If rank* $\begin{bmatrix} \boldsymbol{U}_{0,T,L-T+1} \\ \boldsymbol{Y}_{0,L-T+1} \end{bmatrix} = n + Tm$*, then any length-* T *input/output trajectory of system* (6.1) *can be expressed as* $\begin{bmatrix} \boldsymbol{u}_{[0,T-1]} \\ \boldsymbol{y}_{[0,T-1]} \end{bmatrix} = \begin{bmatrix} \boldsymbol{U}_{0,T,L-T+1} \\ \boldsymbol{Y}_{0,T,L-T+1} \end{bmatrix} \boldsymbol{g}$ *where* $\boldsymbol{g} \in \mathbb{R}^{L-T+1}$.

When the rank condition in Lemma 15 holds, linear combination of the columns of the Hankel matrix is an input/output trajectory of the system. We then generate trajectory of length T using

$$\left[\ \tilde{\boldsymbol{u}}(0);\cdots;\tilde{\boldsymbol{u}}(T-1);\tilde{\boldsymbol{y}}(0);\cdots;\tilde{\boldsymbol{y}}(T-1)\ \right]=\mathcal{H}\boldsymbol{g}. \tag{6.14}$$

For convenience, we adopt the notation $\mathcal{H}_u = U_{0,T,L-T+1}$, $\mathcal{H}_u = Y_{0,T,L-T+1}$ in the following sections. To represent the lines of blocks starting from the time $k = 0,\cdots,T-1$, $\mathcal{H}_u^k$ the same as in (6.7) and $\mathcal{H}_y^k := [y_d(k)\ \cdots\ y_d(L-T+k)]$. We also define the stacked blocks starting from $k_1,\cdots,k_2$ as $\mathcal{H}_y^{k_1:k_2} := \left[H_y^{k_1};H_y^{k_1+1};\cdots;H_y^{k_2}\right]$ and $\mathcal{H}_u^{k_1:k_2} := \left[H_u^{k_1};H_u^{k_1+1};\cdots;H_u^{k_2}\right]$.

6.4.2 Trajectory generation

Using the transition dynamics (6.13), the probability density function of a length-T trajectory is

$$p_{\pi_{\boldsymbol{\theta}}}(\boldsymbol{\tau}) = p\left(\mathcal{X}\left(T_0-1\right)\right)\prod_{k=T_0-1}^{T-1} p\left(\boldsymbol{\theta}\boldsymbol{y}(k)+\boldsymbol{w}(k)|\mathcal{X}(k)\right)p\left(\mathcal{X}(k+1)|\mathcal{X}(k),\boldsymbol{\theta}\boldsymbol{y}(k)+\boldsymbol{w}(k)\right),$$

which is uniquely determined by $\mathcal{X}\left(T_0-1\right)$ and the sequences $\boldsymbol{w}(T_0-1),\cdots,\boldsymbol{w}(T-1)$.

In analogy with the derivation in (6.9), we aim to generate the trajectory starting from $\mathcal{X}\left(T_0-1\right)$ under perturbations on actions given by $\boldsymbol{w}(T_0-1),\cdots,\boldsymbol{w}(T-1)$. From the control policy $\boldsymbol{u}(k) = \boldsymbol{\theta}\boldsymbol{y}(k)+\boldsymbol{w}(k)$, the trajectory subject to the perturbations $\boldsymbol{w}(T_0),\cdots,\boldsymbol{w}(T-1)$ should satisfy

$$\left[\mathcal{H}_u^{T_0-1:T-1}\right]\boldsymbol{g} = \left(\boldsymbol{I}_{T-T_0}\otimes\boldsymbol{\theta}\right)\left[\mathcal{H}_y^{T_0-1:T-1}\right]\boldsymbol{g}+\begin{bmatrix}\boldsymbol{w}(T_0-1)\\ \vdots\\ \boldsymbol{w}(T-1)\end{bmatrix}. \tag{6.15}$$

Note that the generated extended initial state is given by $\tilde{\mathcal{X}}\left(T_0-1\right) := \left[\mathcal{H}_y^{0:T_0-1};\mathcal{H}_u^{0:T_0-2}\right]\boldsymbol{g}$. Together with the constraints in (6.15) gives

$$\underbrace{\begin{bmatrix}\mathcal{H}_u^{T_0-1:T-1}-\left(\boldsymbol{I}_{T-T_0}\otimes\boldsymbol{\theta}\right)\mathcal{H}_y^{T_0-1:T-1}\\ \hline \mathcal{H}_y^{0:T_0-1}\\ \mathcal{H}_u^{0:T_0-2}\end{bmatrix}}_{G_\theta}\boldsymbol{g} = \underbrace{\begin{bmatrix}\boldsymbol{w}(T_0-1)\\ \vdots\\ \boldsymbol{w}(T-1)\\ \hline \mathcal{X}(T_0-1)\end{bmatrix}}_{\boldsymbol{R}} \tag{6.16}$$

Note that the matrix $\boldsymbol{G_\theta} \in \mathbb{R}^{(Tm+T_0d)\times(L-T+1)}$ is not a square matrix and there might be multiple solutions to (6.16). Moreover, $\boldsymbol{G_\theta}$ may not be full row rank and thus $\left(\boldsymbol{G_\theta}\boldsymbol{G_\theta}^\top\right)$ may not be invertible. Here, we compute the eigenvalue decomposition of $\left(\boldsymbol{G_\theta}\boldsymbol{G_\theta}^\top\right)$. Let s be the number of nonzero eigenvalue of $\left(\boldsymbol{G_\theta}\boldsymbol{G_\theta}^\top\right)$. Let λ_i be the i-th non-zero eigenvalue and $\boldsymbol{p}_i$ be the associated eigenvector of $\left(\boldsymbol{G_\theta}\boldsymbol{G_\theta}^\top\right)$. Denote $\boldsymbol{P_\theta} := \begin{bmatrix}\boldsymbol{p}_1 & \boldsymbol{p}_2 & \cdots & \boldsymbol{p}_s\end{bmatrix}$ and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_s)$. Then clearly $\left(\boldsymbol{G_\theta}\boldsymbol{G_\theta}^\top\right) = \boldsymbol{P_\theta}\boldsymbol{\Lambda}\boldsymbol{P_\theta}^\top$ and we compute the solution of (6.16) given by

$$\boldsymbol{g}^* = \boldsymbol{G_\theta}^\top \boldsymbol{P_\theta}\boldsymbol{\Lambda}^{-1}\boldsymbol{P_\theta}^\top \left[\ \boldsymbol{w}(T_0-1); \cdots; \boldsymbol{w}(T-1); \mathcal{X}(T_0-1)\ \right]. \tag{6.17}$$

Next, we prove the existence and uniqueness of the trajectory generated by (6.16). The goal is to show that given $(\boldsymbol{w}(T_0-1), \cdots, \boldsymbol{w}(T-1), \mathcal{X}(T_0-1))$, any $\boldsymbol{g}$ that satisfies (6.16) will generate the same trajectory using $\mathcal{H}\boldsymbol{g}$. So it is suffice to choose the closed-form solution in (6.17).

Theorem 9. *If rank*$(\mathcal{H}) = n + Tm$*, there exists at least one solution* $\boldsymbol{g}^*$ *such that* (6.16) *holds. Given* $(\boldsymbol{w}(T_0-1), \cdots, \boldsymbol{w}(T-1), \mathcal{X}(T_0-1))$ *and any* $\boldsymbol{g}$ *that solves* (6.16)*,* $\mathcal{H}\boldsymbol{g}^*$ *generates the same unique trajectory under the control policy* (6.4) *parameterized by* $\boldsymbol{\theta}$*.*

The proof of Theorem 9 is not as straightforward as Theorem 8, because $\boldsymbol{G_\theta}$ and $\boldsymbol{R}$ in (6.16) may not be full row-rank. The detailed proof is given in Appendix C.5 and we sketch the proof as follows. We use the mapping from $\boldsymbol{x}(0)$ to $\mathcal{X}(T_0-1)$ to show that the rank of $\boldsymbol{R}$ in (6.16) is at most $n + Tm$. Leveraging the relation in every T_0 blocks derived from (6.13), we show in Lemma 16 that rank$(\boldsymbol{G_\theta}) = n + Tm$ if rank$(\mathcal{H}) = n + Tm$. The existence of a solution in (6.16) is therefore guaranteed by the same row-rank of the two sides. The uniqueness of the trajectory generated by $\mathcal{H}\boldsymbol{g}^*$ is proved by showing that the Null space $\mathcal{N}(\boldsymbol{G_\theta})$ is the same as $\mathcal{N}(\mathcal{H})$.

We can generate a trajectory $\mathcal{H}\boldsymbol{g}^*$ by randomly sampling $\mathcal{X}(T_0-1)$ and $\boldsymbol{w}(t) \in \mathcal{D}$ for $t = T_0-1, \cdots, T-1$. For the cost (6.2) calculated on the trajectory of the length K, we setup $T = K + T_0 - 1$, and using the generated trajectory $\mathcal{H}\boldsymbol{g}^*$ from T_0-1 to $T-1$ to train the controller. The detailed algorithm can be found in Appendix C.6.

6.5 Experiment

We end the chapter with case studies for the voltage control of the power distribution system in [6]. Both state-feedback and output-feedback control are studied in these systems. Code is available at `https://github.com/Wenqi-Cui/Trajectory-Generation`.

6.5.1 Experiment Setup.

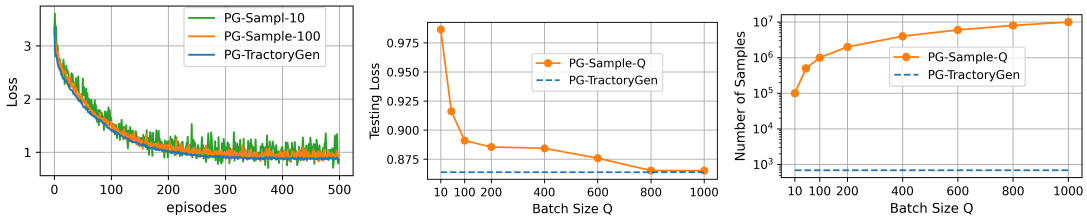
We use REINFORCE policy gradient algorithm [53] to train a linear feedback controller, with the goal to minimize the cost of trajectories with length K. Let E be the number of episode in training and Q be the batch size of trajectories collected for each episode, respectively. Standard policy gradient algorithms needs $Q \times K \times E$ samples. We compare the performance of the REINFORCE policy gradient algorithm using the generated trajectories (labeled as PG-TrajectoryGen) and the same algorithm using trajectories sampled by interacting with the system (labeled as PG-Sample-Q for the batchsize Q). For testing, we randomly sample 800 initial states and compare the cost on trajectories starting from these states.

We conduct experiments on the voltage control problem in IEEE 33bus test feeder [6, 105], where $\boldsymbol{x}(t) \in \mathbb{R}^{32}$. We adopt the Lindisflow model where the dynamics of voltage is described by a linear transition model [6, 89]. The state $\boldsymbol{x}(t) \in \mathbb{R}^{32}$ is voltage in all the buses apart from the reference bus (the voltage of the reference bus is fixed). The action is the reactive power in each bus. We assume that there is no real-time communication between buses during real-time implementation, so the action at each bus can only change with the local measurement of voltage. The goal is to train a linear decentralized feedback controller to minimize total voltage deviation as well as the control effort in the time horizon $K = 20$, written as $J(\boldsymbol{\theta}) = \sum_{k=1}^{K} \|\boldsymbol{y}(k)\|_1 + 0.3 \|\boldsymbol{u}_{\boldsymbol{\theta}}(k)\|_1$. The number of training episode is $E = 500$.

6.5.2 Performances

Case I: the state is directly observed. The state is directly observed so the action $\boldsymbol{u}(t) \in \mathbb{R}^{32}$. We setup $T = K = 20$ and collect historic trajectory of the length $L = (m+1)T - 1 + n = 691$.

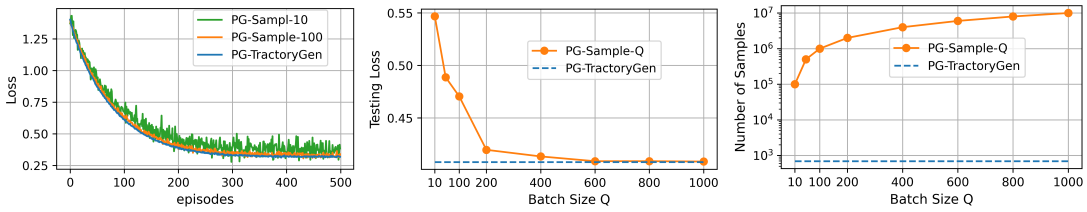
With a sampling period of 1s [124], the data collection takes 691s. In each episode of training, we use Algorithm 3 to generate trajectories with the batchsize 1000. Figure 6.1(a)-(c) compares the training loss, testing loss and the number of samples, respectively. PG-TrajectoryGen achieves the same training and testing loss as PG-Sample-1000 with much smaller number of samples on the system.



(a) Training Loss (b) Testing Loss (c) Number of samples

Figure 6.1: Performance of learning state-feedback controllers in power distribution network. PG-TrajectoryGen achieves the same training and testing loss as PG-Sample-1000 with much smaller number of samples on the system.

Case II: only 20 **elements in the state is observed.** We assume only 20 buses are measured and controlled, so $\boldsymbol{y}(t), \boldsymbol{u}(t) \in \mathbb{R}^{20}$. The observability matrix $\mathcal{O}_{[0,T_0]}(\boldsymbol{A}, \boldsymbol{C})$ becomes full column rank when $T_0 = 3$. The time horizon of trajectory is $K = 20$. According to the trajectory generation algorithm developed in Subsection 6.4.2, we setup $T = K + T_0 - 1 = 22$ and collect historic trajectory of length $L = (m+1)T - 1 + n = 493$. With a sampling period of 1s [124], the data collection takes 493s. Figure 6.2(a)-(c) compares the training loss, testing loss and the number of samples, respectively. PG-TrajectoryGen achieves the same training and testing loss as PG-Sample-1000 with much smaller number of samples on the system.



(a) Training Loss (b) Testing Loss (c) Number of samples

Figure 6.2: Performance of learning output-feedback controllers in power distribution network. PG-TrajectoryGen achieves the same loss as PG-Sample-1000 with much less samples.

6.6 Conclusion

This chapter proposes a trajectory generation algorithm for learning linear feedback controllers in linear systems. We prove that the algorithm generates trajectories with the exact distribution as if they are sampled by interacting with the real system using the updated control policy. In particular, the algorithm extends to systems where the states are not directly observed. This is done by equivalently defining system transition dynamics using input-output trajectories. Experiments show that the proposed method significantly reduces the number of sampled data needed for RL algorithms.

Chapter 7

CONCLUSIONS

To mitigate climate change, power and power systems are undergoing a significant structural transformation to integrate renewables, facilitate decarbonization, and electrify our economies. The electric grid therefore needs to adapt and serve a larger system that is becoming more distributed, having less inertia, and facing more uncertainties. At the same time, the increased controllability of hardware, together with the explosion of available data, offers exciting opportunities to deploy advanced methods that can reshape the landscape of power systems.

This thesis centers around control and machine learning methods to attain sustainable and efficient operation of power systems. We have developed theoretical and algorithmic frameworks for control and machine learning methods that apply to largescale and nonlinear power systems. In Part I, we show how to bridge the gap between learning and safety-critical constraints through structured neural networks guided by control theory and the physics of power systems. Using Lyapunov theory, we extract stabilizing controller structures for transient stability problems, and parameterize the structures by neural networks. To further enforce steady-state tracking and economic dispatch, we propose Neural-PI control, which achieves steady-state optimality distributedly through consensus over neighbors. In addition, we propose a modular approach for transient stability analysis with lossy transmission lines. This provides a simple yet effective approach to optimize control efforts with guaranteed stability regions.

The structured approach for learning-based control provides end-to-end guarantees that are independent of the learning process. In Part II, we further show how those end-to-end guarantees lead to more flexible learning algorithm design, including decentralized learning and trajectory generation algorithms. To relieve the burden of centralized coordination, we propose a decentralized safe learning approach to trains local neural network controller at each node in a model-free setting. To

conduct faster time-scale learning, we develop a sample-efficient algorithm by generating trajectories using past input-to-output data. The key idea is to identify the basis functions that span all possible trajectories, and then leverage the linear combinations of the basis to generate trajectories. By strategically conducting linear combinations, the algorithm adapts to the distributional shift of trajectories resulting from updated control policies.

7.1 Future Directions

The thesis explores the frameworks of combining control theory, power system engineering, and AI techniques to attain sustainable and efficient operation of power systems. Some important future directions include:

1. *Structured Control for Efficient and Safe Operation of Power Systems.* The electric grid naturally yields a hierarchical architecture that tightly integrates physical, control and cyber layers, which makes it tractable to control and optimize the system in a multi-timescale manner. With more distributed energy resources entering the electric grid, how could we adjust this hierarchical architecture to accommodate the distributed power provisions and their uncertainties becomes an important question. In particular, the system should respect the physics of these resources while maintaining stability after plug-ins and plug-outs of devices and topology changes. One possible direction is to start from our previous work on the modular approach for transient stability analysis of networked microgrids [5]. For more complex systems with heterogeneous resources, how do we design the abstraction of modules to respect their physical constraints (e.g., the state-of-charge of energy storage and inverter dynamics) and how can we enforce the safety constraints of these resources are the key challenges to address in the next steps.

2. *Multi-Agent Decentralized Learning and Convergence Analysis.* For the control of power distribution systems with limited communications, our preliminary work in [6] proposed a decentralized safe learning approach that trains local neural network controller at each bus in a model-free setting. It shows that the structure of stabilizing controllers plays a vital role for

the decentralized training to converge without the need for real-time communications. By reformulating the problem in a non-cooperative game setting, we showed that the convexity of the stabilizing structure plays a vital role in the existence of Nash equilibrium [7]. The convergence of decentralized algorithms to a (unique) Nash equilibrium and generalizing the proposed methods to nonlinear power flow models are important future directions for us.

3. *Adaptation to Time-Varying Uncertainties.* Because of the intermittent and uncertain nature of renewable resources, power systems are experiencing larger and faster fluctuations in both generation and load. Existing methods typically represent model mismatch and time variations as bounded noises, and design robust controllers for the worst-case uncertainties. However, as the intermittency and uncertainties grow, these approaches tend to lead to very conservative results. Since the time-varying patterns are typically reflected in the operational data of the system, the massive increases in real-time data can assist in adapting to these uncertainties [125]. The bottleneck lies in safely and effectively leveraging the operational data for online control and decision-making processes. As an initial step, we propose to integrate predictions into the design of adaptive controllers such that the real-time response of the system can be anticipatory to time-varying uncertainties [126]. We design integral law to compensate for deficiencies not captured in predictions, which can be seamlessly combined with most existing controllers (including conventional droop control and emerging neural network-based controllers). Future directions include rigorous analysis of the performance guarantee subject to noises and prediction errors, and case studies using predictions from real-world measurement data.

4. *Sample Efficient Learning Algorithms.* In the context of optimizing controllers in a time-varying environment, updating control laws online is a natural solution. Since the data and interactions during online operations are typically limited, carefully designing data-efficient algorithms to update the controllers is critical. In [8], we have achieved some preliminary results on sample-efficient learning algorithms through trajectory generation. This provides a powerful paradigm that can be envisioned as a "generative model" for learning-based control.

Since new input-output measurements are available when we implement the controller, for more general nonlinear systems, we can think of this as a type of successive linearization. We will further derive the theoretical bound and performance guarantees of the online learning framework through trajectory generation.

BIBLIOGRAPHY

[1] G. L. Barbose, "Us renewables portfolio standards 2021 status update: Early release," Lawrence Berkeley National Laboratory (LBNL), Berkeley, CA (United States), Tech. Rep., 2021.

[2] W. Cui, Y. Jiang, and B. Zhang, "Reinforcement learning for optimal primary frequency control: A lyapunov approach," *IEEE Transactions on Power Systems*, vol. 38, no. 2, pp. 1676–1688, 2023.

[3] W. Cui, Y. Jiang, B. Zhang, and Y. Shi, "Structured neural-pi control with end-to-end stability and output tracking guarantees," *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[4] Y. Jiang, W. Cui, B. Zhang, and J. Cortés, "Stable reinforcement learning for optimal frequency control: A distributed averaging-based integral approach," *IEEE Open Journal of Control Systems*, vol. 1, pp. 194–209, 2022.

[5] W. Cui and B. Zhang, "Equilibrium-independent stability analysis for distribution systems with lossy transmission lines," *IEEE Control Systems Letters*, vol. 6, pp. 3349–3354, 2022.

[6] W. Cui, J. Li, and B. Zhang, "Decentralized safe reinforcement learning for inverter-based voltage control," *Electric Power Systems Research*, vol. 211, p. 108609, 2022.

[7] Y. Jiang, W. Cui, B. Zhang, and J. Cortés, "Equilibria of fully decentralized learning in networked systems," *Learning for Dynamics and Control Conference (L4DC)*, pp. 333–345, 2023.

[8] W. Cui, L. Huang, W. Yang, and B. Zhang, "Efficient reinforcement learning through trajectory generation," *Learning for Dynamics and Control Conference (L4DC)*, pp. 371–382, 2023.

[9] B. Kroposki, B. Johnson, Y. Zhang, V. Gevorgian, P. Denholm, B.-M. Hodge, and B. Hannegan, "Achieving a 100% renewable grid: Operating electric power systems with extremely high levels of variable renewable energy," *IEEE Power and Energy Magazine*, vol. 15, no. 2, pp. 61–73, 2017.

[10] P. Kundur, N. J. Balu, and M. G. Lauby, *Power system stability and control*. McGraw-hill New York, 1994, vol. 7.

[11] B. K. Poolla, S. Bolognani, and F. Dorfler, "Optimal placement of virtual inertia in power grids," *IEEE Transactions on Automatic Control*, 2017.

[12] Z. Zhang, E. Du, F. Teng, N. Zhang, and C. Kang, "Modeling frequency dynamics in unit commitment with a high share of renewable energy," *IEEE Transactions on Power Systems*, 2020.

[13] C. Zhao, U. Topcu, N. Li, and S. Low, "Design and stability of load-side primary frequency control in power systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1177–1189, 2014.

[14] E. Mallada, C. Zhao, and S. Low, "Optimal load-side control for frequency regulation in smart grids," *IEEE Transactions on Automatic Control*, vol. 62, no. 12, pp. 6294–6309, 2017.

[15] A. Ademola-Idowu and B. Zhang, "Frequency stability using inverter power control in low-inertia power systems," *IEEE Transactions on Power Systems*, pp. 1–1, 2020.

[16] B. B. Johnson, S. V. Dhople, A. O. Hamadeh, and P. T. Krein, "Synchronization of parallel single-phase inverters with virtual oscillator control," *IEEE Transactions on Power Electronics*, vol. 29, no. 11, pp. 6124–6138, 2013.

[17] O. Stanojev, U. Markovic, P. Aristidou, G. Hug, D. S. Callaway, and E. Vrettos, "Mpc-based fast frequency control of voltage source converters in low-inertia power systems," *IEEE Transactions on Power Systems*, pp. 1–1, 2020.

[18] X. Chen, G. Qu, Y. Tang, S. Low, and N. Li, "Reinforcement learning for decision-making and control in power systems: Tutorial, review, and vision," *arXiv preprint arXiv:2102.01168*, 2021.

[19] Z. Yan and Y. Xu, "Data-driven load frequency control for stochastic power systems: A deep reinforcement learning method with continuous action search," *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 1653–1656, 2018.

[20] G. Qu, A. Wierman, and N. Li, "Scalable reinforcement learning of localized policies for multi-agent networked systems," in *Learning for Dynamics and Control*. PMLR, 2020, pp. 256–266.

[21] Q. Huang, R. Huang, W. Hao, J. Tan, R. Fan, and Z. Huang, "Adaptive power system emergency control using deep reinforcement learning," *IEEE Transactions on Smart Grid*, 2019.

[22] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[23] Y. Zhang and J. Cortés, "Distributed transient frequency control for power networks with stability and performance guarantees," *Automatica*, vol. 105, pp. 274–285, 2019.

[24] P. W. Sauer, M. A. Pai, and J. H. Chow, *Power system dynamics and stability: with synchrophasor measurement and power system toolbox*. John Wiley & Sons, 2017.

[25] A. D. Domínguez-García, "Models for impact assessment of wind-based power generation on frequency control," in *Control and Optimization Methods for Electric Smart Grids*. Springer, 2012, pp. 149–165.

[26] B. Xu, Y. Shi, D. S. Kirschen, and B. Zhang, "Optimal battery participation in frequency regulation markets," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6715–6725, 2018.

[27] P. Hidalgo-Gonzalez, R. Henriquez-Auba, D. S. Callaway, and C. J. Tomlin, "Frequency regulation using data-driven controllers in power grids with variable inertia due to renewable energy," in *2019 IEEE Power Energy Society General Meeting (PESGM)*, 2019, pp. 1–5.

[28] Y. Jiang, E. Cohn, P. Vorobev, and E. Mallada, "Storage-based frequency shaping control," *IEEE Transactions on Power Systems*, 2021.

[29] J. H. Chow and K. W. Cheung, "A toolbox for power system dynamics and control engineering education and research," *IEEE transactions on Power Systems*, vol. 7, no. 4, pp. 1559–1564, 1992.

[30] D. Tabas and B. Zhang, "Optimal l-infinity frequency control in microgrids considering actuator saturation," *arXiv preprint arXiv:1910.03720*, 2019.

[31] Y. Jiang, R. Pates, and E. Mallada, "Dynamic droop control in low-inertia power systems," *IEEE Transactions on Automatic Control*, 2020.

[32] E. Weitenberg, Y. Jiang, C. Zhao, E. Mallada, C. De Persis, and F. Dörfler, "Robust decentralized secondary frequency control in power systems: Merits and tradeoffs," *IEEE Transactions on Automatic Control*, vol. 64, no. 10, pp. 3967–3982, Oct. 2019.

[33] E. Weitenberg, C. De Persis, and N. Monshizadeh, "Exponential convergence under distributed averaging integral frequency control," *Automatica*, vol. 98, pp. 103–113, Dec. 2018.

[34] S. Sastry, *Nonlinear systems: analysis, stability, and control*. Springer Science & Business Media, 2013, vol. 10.

[35] A. Arapostathis, S. Sastry, and P. Varaiya, "Global analysis of swing dynamics," *IEEE Transactions on Circuits and Systems*, vol. 29, no. 10, pp. 673–679, 1982.

[36] A. Griewank, "On automatic differentiation," *Mathematical Programming: recent developments and applications*, vol. 6, no. 6, pp. 83–107, 1989.

[37] A. Ortega and F. Milano, "Generalized model of vsc-based energy storage systems for transient stability analysis," *IEEE transactions on Power Systems*, vol. 31, no. 5, pp. 3369–3380, 2015.

[38] P. Demetriou, M. Asprou, J. Quiros-Tortos, and E. Kyriakides, "Dynamic ieee test systems for transient analysis," *IEEE Systems Journal*, vol. 11, no. 4, pp. 2108–2117, 2015.

[39] T. Nishikawa and A. E. Motter, "Comparative analysis of existing models for power-grid synchronization," *New Journal of Physics*, vol. 17, no. 1, p. 015012, 2015.

[40] D. S. Kirschen and G. Strbac, *Fundamentals of Power System Economics*, 2nd ed. Wiley, 2019.

[41] J. H. Eto, J. Undrill, P. Mackin, R. Daschmans, B. Williams, H. Illian, C. Martinez, M. O'Malley, K. Coughlin, and K. H. LaCommare, "Use of frequency response metrics to assess the planning and operating requirements for reliable integration of variable renewable generation," Lawrence Berkeley National Laboratory, Tech. Rep., 2010.

[42] B. Kroposki, B. Johnson, Y. Zhang, V. Gevorgian, P. Denholm, B. Hodge, and B. Hannegan, "Achieving a 100% renewable grid: Operating electric power systems with extremely high levels of variable renewable energy," *IEEE Power and Energy Magazine*, vol. 15, no. 2, pp. 61–73, Mar. 2017.

[43] S. Nalley and A. LaRose, "Annual energy outlook 2021 with projections to 2050," U.S. Energy Information Administration, Tech. Rep., 2021.

[44] L. Bird, M. Milligan, and D. Lew, "Integrating variable renewable energy: Challenges and solutions," National Renewable Energy Laboratory, Tech. Rep., 2013.

[45] C. Zhao, E. Mallada, and F. Dörfler, "Distributed frequency control for stability and economic dispatch in power networks," in *Proc. of American Control Conference*, July 2015, pp. 2359–2364.

[46] N. Li, C. Zhao, and L. Chen, "Connecting automatic generation control and economic dispatch from an optimization view," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 3, pp. 254–264, Sept. 2016.

[47] F. Dörfler and S. Grammatico, "Gather-and-broadcast frequency control in power systems," *Automatica*, vol. 79, pp. 296–305, May 2017.

[48] J. Schiffer, F. Dörfler, and E. Fridman, "Robustness of distributed averaging control in power systems: Time delays & dynamic communication topology," *Automatica*, vol. 80, pp. 261–271, June 2017.

[49] Z. Wang, F. Liu, S. H. Low, C. Zhao, and S. Mei, "Distributed frequency control with operational constraints, part II: Network power balance," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 53–64, Jan. 2019.

[50] P. You, Y. Jiang, E. Yeung, D. F. Gayme, and E. Mallada, "On the stability, economic efficiency and incentive compatibility of electricity market dynamics," 2021.

[51] A. Cherukuri, T. Stegink, C. D. Persis, A. J. van der Schaft, and J. Cortés, "Frequency-driven market mechanisms for optimal dispatch in power networks," *Automatica*, vol. 133, p. 109861, Nov. 2021.

[52] J. Schiffer and F. Dörfler, "On stability of a distributed averaging PI frequency and active power controlled differential-algebraic power system model," in *Proc. of European Control Conference*, June 2016, pp. 1487–1492.

[53] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. The MIT Press, 2018.

[54] H. Yin, P. Liu, K. Liu, L. Cao, L. Zhang, Y. Gao, and X. Hei, "Ns3-ai: Fostering artificial intelligence algorithms for networking research," in *Proceedings of the 2020 Workshop on Ns-3*, ser. WNS3 '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 57–64. [Online]. Available: https://doi.org/10.1145/3389400.3389404

[55] D. Ernst, M. Glavic, and L. Wehenkel, "Power systems stability control: reinforcement learning framework," *IEEE Transactions on Power Systems*, vol. 19, no. 1, pp. 427–435, Feb. 2004.

[56] X. Chen, G. Qu, Y. Tang, S. Low, and N. Li, "Reinforcement learning for selective key applications in power systems: Recent advances and future challenges," *IEEE Transactions on Smart Grid (early access)*, 2022.

[57] D. Ye, M. Zhang, and D. Sutanto, "A hybrid multiagent framework with Q-learning for power grid systems restoration," *IEEE Transactions on Power Systems*, vol. 26, no. 4, pp. 2434–2441, Nov. 2011.

[58] V. P. Singh, N. Kishor, and P. Samuel, "Distributed multi-agent system-based load frequency control for multi-area power system in smart grid," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 5151–5160, June 2017.

[59] L. Zhang, H. Yin, Z. Zhou, S. Roy, and Y. Sun, "Enhancing wifi multiple access performance with federated deep reinforcement learning," in *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, 2020, pp. 1–6.

[60] Z. Yan and Y. Xu, "A multi-agent deep reinforcement learning method for cooperative load frequency control of a multi-area power system," *IEEE Transactions on Power Systems*, vol. 35, no. 6, pp. 4599–4608, Nov. 2020.

[61] C. Chen, M. Cui, F. Li, S. Yin, and X. Wang, "Model-free emergency frequency control based on reinforcement learning," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2336–2346, Apr. 2021.

[62] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[63] M. Andreasson, D. V. Dimarogonas, H. Sandberg, and K. H. Johansson, "Distributed control of networked dynamical systems: Static feedback, integral action and consensus," *IEEE Transactions on Automatic Control*, vol. 59, no. 7, pp. 1750–1764, 2014.

[64] J.-B. Hiriart-Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms I: Fundamentals*. Springer Science & Business Media, 1996, vol. 305.

[65] T. Athay, R. Podmore, and S. Virmani, "A practical method for the direct analysis of transient stability," *IEEE Transactions on Power Apparatus and Systems*, no. 2, pp. 573–584, 1979.

[66] W. Cui, Y. Jiang, B. Zhang, and Y. Shi, "Structured neural-pi control for networked systems: Stability and steady-state optimality guarantees," *arXiv preprint arXiv:2206.00261*, 2022.

[67] H. Xu, A. D. Domínguez-García, V. V. Veeravalli, and P. W. Sauer, "Data-driven voltage regulation in radial power distribution systems," *IEEE Transactions on Power Systems*, vol. 35, no. 3, pp. 2133–2143, 2019.

[68] S. C. Ross and J. L. Mathieu, "A method for ensuring a load aggregator's power deviations are safe for distribution networks," *Electric Power Systems Research*, vol. 189, p. 106781, 2020.

[69] Y. Zhang and L. Xie, "A transient stability assessment framework in power electronic-interfaced distribution systems," *IEEE Transactions on Power Systems*, vol. 31, no. 6, pp. 5106–5114, 2016.

[70] H.-D. Chiang, "Study of the existence of energy functions for power systems with losses," *IEEE Transactions on Circuits and Systems*, vol. 36, no. 11, pp. 1423–1429, 1989.

[71] ——, *Direct methods for stability analysis of electric power systems: theoretical foundation, BCU methodologies, and applications*. John Wiley & Sons, 2011.

[72] W. H. Kersting, "Radial distribution test feeders," *IEEE Transactions on Power Systems*, vol. 6, no. 3, pp. 975–985, 1991.

[73] B. A. Robbins, C. N. Hadjicostis, and A. D. Domínguez-García, "A two-stage distributed architecture for voltage control in power distribution systems," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1470–1482, 2012.

[74] B. Zhang, A. Y. Lam, A. D. Domínguez-García, and D. Tse, "An optimal and distributed method for voltage regulation in power distribution systems," *IEEE Transactions on Power Systems*, vol. 30, no. 4, pp. 1714–1726, 2015.

[75] T. Huang, S. Gao, and L. Xie, "A neural lyapunov approach to transient stability assessment of power electronics-interfaced networked microgrids," *IEEE Transactions on Smart Grid*, vol. 13, no. 1, pp. 106–118, 2021.

[76] T. L. Vu and K. Turitsyn, "Lyapunov functions family approach to transient stability assessment," *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1269–1277, 2015.

[77] W. Cui and B. Zhang, "Lyapunov-regularized reinforcement learning for power system transient stability," *IEEE Control Systems Letters*, vol. 6, pp. 974–979, 2021.

[78] X. Miao and M. D. Ilić, "Modeling and distributed control of microgrids: A negative feedback approach," in *CDC*, 2019.

[79] G. H. Hines, M. Arcak, and A. K. Packard, "Equilibrium-independent passivity: A new definition and numerical certification," *Automatica*, vol. 47, no. 9, pp. 1949–1956, 2011.

[80] M. Arcak, C. Meissen, and A. Packard, *Networks of dissipative systems: compositional certification of stability, performance, and safety*. Springer, 2016.

[81] B. B. Johnson, M. Sinha, N. G. Ainsworth, F. Dörfler, and S. V. Dhople, "Synthesizing virtual oscillators to control islanded inverters," *IEEE Transactions on Power Electronics*, vol. 31, no. 8, pp. 6002–6015, 2015.

[82] M. Davis, "Solar market insight report," *Wood Mackenzie Power and Renewables*, 2021.

[83] K. Turitsyn, P. Sulc, S. Backhaus, and M. Chertkov, "Options for control of reactive power by distributed photovoltaic generators," *Proceedings of the IEEE*, vol. 99, no. 6, pp. 1063–1073, 2011.

[84] H.-G. Yeh, D. F. Gayme, and S. H. Low, "Adaptive var control for distribution circuits with photovoltaic generators," *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1656–1663, 2012.

[85] B. Zhang, A. Y. Lam, A. D. Domínguez-García, and D. Tse, "An optimal and distributed method for voltage regulation in power distribution systems," *IEEE Transactions on Power Systems*, vol. 30, no. 4, pp. 1714–1726, 2014.

[86] N. Li, G. Qu, and M. Dahleh, "Real-time decentralized voltage control in distribution networks," in *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2014, pp. 582–588.

[87] S. Bolognani and S. Zampieri, "A distributed control strategy for reactive power compensation in smart microgrids," *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2818–2833, 2013.

[88] J. Feng, Y. Shi, G. Qu, S. H. Low, A. Anandkumar, and A. Wierman, "Stability constrained reinforcement learning for decentralized real-time voltage control," *IEEE Transactions on Control of Network Systems*, 2023.

[89] H. Zhu and H. J. Liu, "Fast local voltage control under limited reactive power: Optimality and stability analysis," *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 3794–3803, 2015.

[90] G. Qu and N. Li, "Optimal distributed feedback voltage control under limited reactive power," *IEEE Transactions on Power Systems*, vol. 35, no. 1, pp. 315–331, 2019.

[91] Y.-Y. Hsu and F.-C. Lu, "A combined artificial neural network-fuzzy dynamic programming approach to reactive power/voltage control in a distribution substation," *IEEE transactions on Power Systems*, vol. 13, no. 4, pp. 1265–1271, 1998.

[92] S. Toma, T. Senjyu, Y. Miyazato, A. Yona, K. Tanaka, and C.-H. Kim, "Decentralized voltage control in distribution system using neural network," in *2008 IEEE 2nd International Power and Energy Conference*. IEEE, 2008, pp. 1557–1562.

[93] X. Shen, H. Wang, J. Li, Q. Su, and L. Gao, "Distributed secondary voltage control of islanded microgrids based on rbf-neural-network sliding-mode technique," *IEEE Access*, vol. 7, pp. 65 616–65 623, 2019.

[94] Q. Yang, G. Wang, A. Sadeghi, G. B. Giannakis, and J. Sun, "Two-timescale voltage control in distribution grids using deep reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2313–2323, 2019.

[95] Y. Gao, W. Wang, and N. Yu, "Consensus multi-agent reinforcement learning for volt-var control in power distribution networks," *IEEE Transactions on Smart Grid*, 2021.

[96] J. Duan, D. Shi, R. Diao, H. Li, Z. Wang, B. Zhang, D. Bian, and Z. Yi, "Deep-reinforcement-learning-based autonomous voltage control for power grid operations," *IEEE Transactions on Power Systems*, vol. 35, no. 1, pp. 814–817, 2019.

[97] N. E. M. Association *et al.*, *American National Standard for Electric Power Systems and Equipment-Voltage Ratings (60 Hertz)*. National Electrical Manufacturers Association, 1996.

[98] A. Vaccaro, G. Velotto, and A. F. Zobaa, "A decentralized and cooperative architecture for optimal voltage regulation in smart grids," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 10, pp. 4593–4602, 2011.

[99] M. Jafari, T. O. Olowu, and A. I. Sarwat, "Optimal smart inverters volt-var curve selection with a multi-objective volt-var optimization using evolutionary algorithm approach," in *2018 North American Power Symposium (NAPS)*. IEEE, 2018, pp. 1–6.

[100] Y. Shi, B. Xu, D. Wang, and B. Zhang, "Using battery storage for peak shaving and frequency regulation: Joint optimization for superlinear gains," *IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 2882–2894, 2017.

[101] J. Yu, Y. Weng, and R. Rajagopal, "Patopaem: A data-driven parameter and topology joint estimation framework for time-varying system in distribution grids," *IEEE Transactions on Power Systems*, vol. 34, no. 3, pp. 1682–1692, 2018.

[102] J. Zhang, Y. Wang, Y. Weng, and N. Zhang, "Topology identification and line parameter estimation for non-pmu distribution network: A numerical method," *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 4440–4453, 2020.

[103] S. Lin and H. Zhu, "Data-driven modeling for distribution grids under partial observability," *arXiv preprint arXiv:2108.08350*, 2021.

[104] R. Kavet, "Characterization of radiofrequency emissions from two models of wireless smart-meters," *Project Manager Electric Power Research Institute, EPRI*, 2011.

[105] M. E. Baran and F. F. Wu, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Power Engineering Review*, vol. 9, no. 4, pp. 101–102, 1989.

[106] J. Feng, W. Cui, J. Cortés, and Y. Shi, "Bridging transient and steady-state performance in voltage control: A reinforcement learning approach with safe gradient flow," *IEEE Control Systems Letters*, vol. 7, pp. 2845–2850, 2023.

[107] Y. Zheng, L. Furieri, M. Kamgarpour, and N. Li, "Sample complexity of linear quadratic gaussian (lqg) control for output feedback systems," in *Learning for dynamics and control*. PMLR, 2021, pp. 559–570.

[108] B. Hu, K. Zhang, N. Li, M. Mesbahi, M. Fazel, and T. Başar, "Towards a theoretical foundation of policy optimization for learning control policies," *arXiv preprint arXiv:2210.04810*, 2022.

[109] L. Zhang, H. Yin, S. Roy, and L. Cao, "Multiaccess point coordination for next-gen wi-fi networks aided by deep reinforcement learning," *IEEE Systems Journal*, vol. 17, no. 1, pp. 904–915, 2023.

[110] M. Fazel, R. Ge, S. Kakade, and M. Mesbahi, "Global convergence of policy gradient methods for the linear quadratic regulator," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1467–1476.

[111] Y. Jin, Z. Yang, and Z. Wang, "Is pessimism provably efficient for offline rl?" in *International Conference on Machine Learning*. PMLR, 2021, pp. 5084–5096.

[112] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv preprint arXiv:2005.01643*, 2020.

[113] S. Fujimoto and S. S. Gu, "A minimalist approach to offline reinforcement learning," *Advances in neural information processing systems*, vol. 34, pp. 20 132–20 145, 2021.

[114] S. K. S. Ghasemipour, D. Schuurmans, and S. S. Gu, "Emaq: Expected-max q-learning operator for simple yet effective offline and online rl," in *International Conference on Machine Learning*. PMLR, 2021, pp. 3682–3691.

[115] C. Gulcehre, Z. Wang, A. Novikov, T. Paine, S. Gómez, K. Zolna, R. Agarwal, J. S. Merel, D. J. Mankowitz, C. Paduraru *et al.*, "Rl unplugged: A suite of benchmarks for offline reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7248–7259, 2020.

[116] G. Ostrovski, P. S. Castro, and W. Dabney, "The difficulty of passive learning in deep reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 23 283–23 295, 2021.

[117] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. De Moor, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.

[118] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 909–924, 2019.

[119] I. Markovsky and F. Dörfler, "Identifiability in the behavioral setting," *IEEE Transactions on Automatic Control*, 2022.

[120] Y. Tang, Y. Zheng, and N. Li, "Analysis of the optimization landscape of linear quadratic gaussian (lqg) control," in *Learning for Dynamics and Control*. PMLR, 2021, pp. 599–610.

[121] J. P. Hespanha, *Linear systems theory*. Princeton university press, 2009.

[122] C. Jin, S. Kakade, A. Krishnamurthy, and Q. Liu, "Sample-efficient reinforcement learning of undercomplete pomdps," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18 530–18 539, 2020.

[123] L. Huang, J. Zhen, J. Lygeros, and F. Dörfler, "Robust data-enabled predictive control: Tractable formulations and performance guarantees," *arXiv preprint arXiv:2105.07199*, 2021.

[124] B. Chen, P. L. Donti, K. Baker, J. Z. Kolter, and M. Bergés, "Enforcing policy feasibility constraints through differentiable projection for energy optimization," in *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*, 2021, pp. 199–210.

[125] W. Cui, W. Yang, and B. Zhang, "A frequency domain approach to predict power system transients," *IEEE Transactions on Power Systems*, 2023.

[126] W. Cui, G. Shi, Y. Shi, and B. Zhang, "Leveraging predictions in power system frequency control: an adaptive approach," *IEEE Conference on Decision and Control (CDC)*, 2023.

[127] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. Cambridge University Press, 2012.

Appendix A

PROOF FOR CHAPTER 2

A.1 Proof of Lemma 2

Proof. The proof is similar to the one of [32, Lemma 14], which bounds $V(\boldsymbol{\delta}, \boldsymbol{\omega})$ term by term. Firstly, using the Rayleigh-Ritz theorem [127], the kinetic energy term, $\frac{1}{2}\sum_{i=1}^{n} M_i(\omega_i - \omega^*)^2$, is lower bounded by $\frac{1}{2}\lambda_{\min}(\boldsymbol{M})||\boldsymbol{\omega}-\boldsymbol{\omega}^*||_2^2$ and upper bounded by $\frac{1}{2}\lambda_{\max}(\boldsymbol{M})||\boldsymbol{\omega}-\boldsymbol{\omega}^*||_2^2$. Then, with a direct application of [33, Lemma 4], the potential energy term $W_{\mathrm{p}}(\boldsymbol{\delta})$ in (2.8a) can be bounded by $\beta_1 \left\|\boldsymbol{\delta} - \boldsymbol{\delta}^*\right\|_2^2 \leq W_{\mathrm{p}}(\boldsymbol{\delta}) \leq \beta_2 \left\|\boldsymbol{\delta} - \boldsymbol{\delta}^*\right\|_2^2$ for some constants $\beta_1 > 0$ and $\beta_2 > 0$.

To deal with the cross term $W_{\mathrm{c}}(\boldsymbol{\delta})$, we define $p_{\mathrm{e},i}(\boldsymbol{\delta}) := \sum_{j=1}^{n} B_{ij}\sin(\delta_{ij})$. Then, $W_{\mathrm{c}}(\boldsymbol{\delta}) = (\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta})-\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*))^T\boldsymbol{M}(\boldsymbol{\omega}-\boldsymbol{\omega}^*)$. Clearly, $-|W_{\mathrm{c}}(\boldsymbol{\delta})| \leq W_{\mathrm{c}}(\boldsymbol{\delta}) \leq |W_{\mathrm{c}}(\boldsymbol{\delta})|$. For $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, $2|\boldsymbol{x}^T\boldsymbol{y}| \leq ||\boldsymbol{x}||_2^2 + ||\boldsymbol{y}||_2^2$. Thus, we have

$$
\begin{aligned}
|W_{\mathrm{c}}(\boldsymbol{\delta})| &\leq \frac{1}{2}\left(||\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta})-\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*)||_2^2 + ||\boldsymbol{M}(\boldsymbol{\omega}-\boldsymbol{\omega}^*)||_2^2\right) \\
&\leq \frac{1}{2}\left(\gamma_2||\boldsymbol{\delta}-\boldsymbol{\delta}^*||_2^2 + \lambda_{\max}(\boldsymbol{M})^2||\boldsymbol{\omega}-\boldsymbol{\omega}^*||_2^2\right) ,
\end{aligned}
$$

where the second inequality comes from [33, Lemma 4] and the Rayleigh-Ritz theorem, with some $\gamma_2 > 0$. Hence, the $W_{\mathrm{c}}(\boldsymbol{\delta})$ term is lower bounded by $-\frac{1}{2}\left(\gamma_2||\boldsymbol{\delta}-\boldsymbol{\delta}^*||_2^2 + \lambda_{\max}(\boldsymbol{M})^2||\boldsymbol{\omega}-\boldsymbol{\omega}^*||_2^2\right)$ and upper bounded by $\frac{1}{2}\left(\gamma_2||\boldsymbol{\delta}-\boldsymbol{\delta}^*||_2^2 + \lambda_{\max}(\boldsymbol{M})^2||\boldsymbol{\omega}-\boldsymbol{\omega}^*||_2^2\right)$. Finally, combining the inequalities, we can bound the entire Lyapunov function $V(\boldsymbol{\delta}, \boldsymbol{\omega})$ in (2.7) with

$$
\begin{aligned}
\alpha_1 &:= \frac{1}{2}\min\left(\lambda_{\min}(\boldsymbol{M}) - \epsilon\lambda_{\max}(\boldsymbol{M})^2, 2\beta_1 - \epsilon\gamma_2\right) > 0\,, \\
\alpha_2 &:= \frac{1}{2}\max\left(\lambda_{\max}(\boldsymbol{M}) + \epsilon\lambda_{\max}(\boldsymbol{M})^2, 2\beta_2 + \epsilon\gamma_2\right) > 0\,,
\end{aligned}
$$

for sufficiently small $\epsilon > 0$. □

A.2 Proof of Lemma 3

Proof. We start by computing the partial derivatives of $V(\boldsymbol{\delta}, \boldsymbol{\omega})$ with respect to each state, i.e.,

$$\begin{aligned}
\frac{\partial V}{\partial \delta_i} &= p_{\mathrm{e},i}(\boldsymbol{\delta}) - p_{\mathrm{e},i}(\boldsymbol{\delta}^*) + \epsilon \sum_{j=1, j\neq i}^{n} B_{ij}\cos(\delta_{ij}) M_i(\omega_i - \omega^*) \\
&\qquad - \epsilon \sum_{j=1, j\neq i}^{n} B_{ij}\cos(\delta_{ij}) M_j(\omega_j - \omega^*)\,, \\
\frac{\partial V}{\partial \omega_i} &= M_i\left[\omega_i - \omega^* + \epsilon\left(p_{\mathrm{e},i}(\boldsymbol{\delta}) - p_{\mathrm{e},i}(\boldsymbol{\delta}^*)\right)\right]\,.
\end{aligned}$$

Therefore, the time derivative of $V(\boldsymbol{\delta}, \boldsymbol{\omega})$, i.e., $\dot{V}(\boldsymbol{\delta}, \boldsymbol{\omega})$, is

$$\begin{aligned}
&\sum_{i=1}^{n}\left(\frac{\partial V}{\partial \delta_i}\dot{\delta}_i + \frac{\partial V}{\partial \omega_i}\dot{\omega}_i\right) \\
&= (\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}) - \boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*) + \epsilon \boldsymbol{H}(\boldsymbol{\delta})\boldsymbol{M}(\boldsymbol{\omega} - \boldsymbol{\omega}^*))^T \left(\boldsymbol{\omega} - \mathbf{1}\frac{\mathbf{1}^T\boldsymbol{\omega}}{n}\right) \\
&\quad + [\boldsymbol{\omega} - \boldsymbol{\omega}^* + \epsilon\left(\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}) - \boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*)\right)]^T (\boldsymbol{p}_{\mathrm{m}} - \boldsymbol{D}\boldsymbol{\omega} - \boldsymbol{u}(\boldsymbol{\omega}) - \boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta})) \\
&\quad \underbrace{+ (\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}) - \boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*) + \epsilon \boldsymbol{H}(\boldsymbol{\delta})\boldsymbol{M}(\boldsymbol{\omega} - \boldsymbol{\omega}^*))^T \left(\mathbf{1}\frac{\mathbf{1}^T\boldsymbol{\omega}}{n} - \mathbf{1}\omega^*\right)}_{=0} \\
&\quad - [\boldsymbol{\omega} - \boldsymbol{\omega}^* + \epsilon(\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}) - \boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*))]^T \underbrace{(\boldsymbol{p}_{\mathrm{m}} - \boldsymbol{D}\boldsymbol{\omega}^* - \boldsymbol{u}(\boldsymbol{\omega}^*) - \boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*))}_{=\mathbf{0}} \\
&= \epsilon\left(\boldsymbol{H}(\boldsymbol{\delta})\boldsymbol{M}(\boldsymbol{\omega} - \boldsymbol{\omega}^*)\right)^T (\boldsymbol{\omega} - \boldsymbol{\omega}^*) - (\boldsymbol{\omega} - \boldsymbol{\omega}^*)^T \boldsymbol{D}(\boldsymbol{\omega} - \boldsymbol{\omega}^*) \\
&\quad - \epsilon\left(\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}) - \boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*)\right)^T (\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}) - \boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*)) \\
&\quad - \epsilon\left(\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}) - \boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*)\right)^T \boldsymbol{D}(\boldsymbol{\omega} - \boldsymbol{\omega}^*) \\
&\quad - [\boldsymbol{\omega} - \boldsymbol{\omega}^* + \epsilon\left(\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}) - \boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*)\right)]^T \left(\boldsymbol{u}(\boldsymbol{\omega}) - \boldsymbol{u}(\boldsymbol{\omega}^*)\right)\,,
\end{aligned}$$

which is exactly (2.9). Note that the extra terms in the second equality are added to construct a quadratic format without affecting the the original value of $\dot{V}(\boldsymbol{\delta}, \boldsymbol{\omega})$ since $\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta})^T\mathbf{1} = \mathbf{0}$, $\boldsymbol{H}(\boldsymbol{\delta})^T\mathbf{1} = \mathbf{0}$, and $\boldsymbol{p}_{\mathrm{m}} - \boldsymbol{D}\boldsymbol{\omega}^* - \boldsymbol{u}(\boldsymbol{\omega}^*) - \boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*) = 0$ by the condition at the equilibrium given in (2.4a).

It remains to show that $\boldsymbol{Q}(\boldsymbol{\delta}) \succ 0$, which follows directly from the fact that the Schur complement of the block $\epsilon\boldsymbol{I}$ in $\boldsymbol{Q}(\boldsymbol{\delta})$ is positive definite: $\boldsymbol{D} - \frac{\epsilon}{2}(\boldsymbol{H}(\boldsymbol{\delta})\boldsymbol{M} + \boldsymbol{M}\boldsymbol{H}(\boldsymbol{\delta})) - \frac{\epsilon}{4}\boldsymbol{D}^2 \succ 0$ for

sufficiently small ϵ. □

A.3 Proof of Theorem 2

Let α bound the magnitude of first derivative of r on $\mathbb{X}$. Define an equispaced grid of points on $\mathbb{X}$, where $\beta = \frac{1}{n}$ is the spacing between grid points along each dimension. Corresponding to each grid interval $[k\beta, (k+1)\beta]$, assign a linear function $y(x) = r(k\beta) + \frac{r((k+1)\beta)-r(k\beta)}{\beta}(x - k\beta)$, where $y(k\beta) = r(k\beta)$ and $y((k+1)\beta) = r((k+1)\beta)$. For all $x \in [k\beta, (k+1)\beta]$, from monotonic property, we have $r(k\beta) \leq r(x) \leq r((k+1)\beta)$ and $r(k\beta) \leq y(x) \leq r((k+1)\beta)$. Therefore, we can bound the approximation error by

$$|y(x) - r(x)| \leq |r((k+1)\beta) - r(k\beta)| \tag{A.2}$$

By mean value theorem, we know that

$$r((k+1)\beta) - r(k\beta) = \beta \frac{\partial r(c)}{\partial x} \tag{A.3}$$

for some point c on the line segment between $k\beta$ and $(k+1)\beta$. Given the assumptions made at the outset, $|\frac{\partial r(c)}{\partial x}|$ is bounded by α and therefore $|y(x) - r(x)|$ can be bounded by $\beta\alpha$.

Further, we show that any piece-wise linear function of $y(x) = r(k\beta) + \frac{r((k+1)\beta)-r(k\beta)}{\beta}(x - k\beta)$ can be represented by the proposed construction (2.13)(2.14). Without loss of generosity, assume that $y(x)$ is the positive part and approximated by $f^+(x)$. Let $b_i^1 = 0$, $q^1 = r(\beta)$ and subsequently $b_i^k = (k-1)\beta$, $\sum_{j=1}^k q^j = \frac{r(k\beta)-r((k-1)\beta)}{\beta}$ for $k = 2, 3, \cdots, n$. Then the construction of $f^+(x)$ through (2.13) is exactly the same as $y(x)$. Therefore, $|f(x) - r(x)|$ can also be bounded by $\beta\alpha$. We take $\beta < \frac{\epsilon}{\alpha}$ to complete the proof.

Appendix B

PROOF FOR CHAPTER 3

B.1 Proof of Lemma 8

Proof. We start by showing that $\boldsymbol{\pi}^I(\boldsymbol{s})^\top \hat{\boldsymbol{c}}\tilde{\boldsymbol{E}}\boldsymbol{\phi}\left(\tilde{\boldsymbol{E}}^\top \nabla \boldsymbol{C}\left(\boldsymbol{\pi}^I(\boldsymbol{s})\right)\right) \geq 0$ with equality holds if and only if $\nabla \boldsymbol{C}(\boldsymbol{\pi}^I(\boldsymbol{s})) \in \mathrm{range}(\mathbb{1}_n)$. Expanding the left side of (3.7) and pre-multiplying $\boldsymbol{\pi}^I(\boldsymbol{s})$ gives

$$
\begin{aligned}
&\left(\boldsymbol{\pi}^I(\boldsymbol{s})\right)^\top \left(\hat{\boldsymbol{c}}\tilde{\boldsymbol{E}}\boldsymbol{\phi}\left(\tilde{\boldsymbol{E}}^\top \nabla \boldsymbol{C}(\boldsymbol{\pi}^I(\boldsymbol{s}))\right)\right) \\
&= \sum_{l=(i,j)\in\tilde{\mathcal{E}}} \phi_l\big(\nabla C_i\big(\pi_i^I(s_i)\big) - \nabla C_j\big(\pi_j^I(s_j)\big)\big)\cdot\big(c_i\pi_i^I(s_i) - c_j\pi_j^I(s_j)\big) \\
&= \sum_{l=(i,j)\in\tilde{\mathcal{E}}} \phi_l\left(\nabla C_{\mathrm{o}}\left(c_i\pi_i^I(s_i)\right) - \nabla C_{\mathrm{o}}\left(c_j\pi_j^I(s_j)\right)\right)\cdot\left(c_i\pi_i^I(s_i) - c_j\pi_j^I(s_j)\right)
\end{aligned}
$$

where the last step follows from the cost function in Assumption 1 that $\nabla C_i(\pi_i^I(s_i)) = \nabla C_{\mathrm{o}}(c_i\pi_i^I(s_i))$ for all $i \in \mathcal{V}$.

Since $C_{\mathrm{o}}(\cdot)$ is strictly convex, its gradient $\nabla C_{\mathrm{o}}(\cdot)$ is strictly increasing [62]. Thus,

$$
\big(\nabla C_{\mathrm{o}}\big(c_i\pi_i^I(s_i)\big) - \nabla C_{\mathrm{o}}\big(c_j\pi_j^I(s_j)\big)\big)\big(c_i\pi_i^I(s_i) - c_j\pi_j^I(s_j)\big) \geq 0 \tag{B.1}
$$

with equality holds if and only if $\nabla C_{\mathrm{o}}\left(c_i\pi_i^I(s_i)\right) = \nabla C_{\mathrm{o}}\left(c_j\pi_j^I(s_j)\right)$.

By Controller Design 1, $\phi_l\left(\nabla C_{\mathrm{o}}(c_i\pi_i^I(s_i)) - \nabla C_{\mathrm{o}}(c_j\pi_j^I(s_j))\right)$ is the same sign with $\nabla C_{\mathrm{o}}(c_i\pi_i^I(s_i)) - \nabla C_{\mathrm{o}}(c_j\pi_j^I(s_j))$. Hence, (B.1) implies

$$
\phi_l\left(\nabla C_{\mathrm{o}}\big(c_i\pi_i^I(s_i)\big) - \nabla C_{\mathrm{o}}\left(c_j\pi_j^I(s_j)\right)\right)\left(c_i\pi_i^I(s_i) - c_j\pi_j^I(s_j)\right) \geq 0
$$

with equality holds if and only if $\nabla C_{\mathrm{o}}\big(c_i\pi_i^I(s_i)\big) = \nabla C_{\mathrm{o}}\big(c_j\pi_j^I(s_j)\big)$. This implies $\nabla C_i(\pi_i^I(s_i)) = \nabla C_j(\pi_j^I(s_j))\ \forall l = (i,j) \in \tilde{\mathcal{E}}$ for cost functions satisfying Assumption 1. Since the graph is connected, we further have $\nabla C_1(\pi^I 1(s_1)) = \ldots = \nabla C_n(\pi^I n(s_n))$, i.e., $\nabla \boldsymbol{C}(\boldsymbol{\pi}^I(\boldsymbol{s})) \in \mathrm{range}(\mathbb{1}_n)$.

Then we prove that $\hat{\boldsymbol{c}}\tilde{\boldsymbol{E}}\boldsymbol{\phi}\left(\tilde{\boldsymbol{E}}^\top \nabla \boldsymbol{C}\left(\boldsymbol{\pi}^I(\boldsymbol{s})\right)\right) = \mathbb{0}_n$ if and only if $\nabla \boldsymbol{C}(\boldsymbol{\pi}^I(\boldsymbol{s})) \in \mathrm{range}(\mathbb{1}_n)$ by showing sufficiency and necessity. If $\nabla C(\boldsymbol{\pi}^I(\boldsymbol{s})) \in \mathrm{range}(\mathbb{1}_n)$, we have $\tilde{\boldsymbol{E}}^\top \nabla \boldsymbol{C}\left(\boldsymbol{\pi}^I(\boldsymbol{s})\right) = \mathbb{0}_n$

and thus $\hat{\boldsymbol{c}}\tilde{\boldsymbol{E}}\boldsymbol{\phi}\left(\tilde{\boldsymbol{E}}^\top\nabla\boldsymbol{C}\left(\boldsymbol{\pi}^I\left(\boldsymbol{s}\right)\right)\right) = \mathbb{0}_n$. On the other hand, if $\hat{\boldsymbol{c}}\tilde{\boldsymbol{E}}\boldsymbol{\phi}\left(\tilde{\boldsymbol{E}}^\top\nabla\boldsymbol{C}\left(\boldsymbol{\pi}^I\left(\boldsymbol{s}\right)\right)\right) = \mathbb{0}_n$, Multiplying both sides by $\left(\boldsymbol{\pi}^I\left(\boldsymbol{s}\right)\right)^\top$ gives $\boldsymbol{\pi}^I(\boldsymbol{s})^\top\hat{\boldsymbol{c}}\tilde{\boldsymbol{E}}\boldsymbol{\phi}\left(\tilde{\boldsymbol{E}}^\top\nabla\boldsymbol{C}\left(\boldsymbol{\pi}^I\left(\boldsymbol{s}\right)\right)\right) = 0$ and therefore $\nabla C(\boldsymbol{\pi}^I(\boldsymbol{s})) \in \mathrm{range}(\mathbb{1}_n)$. Hence, $\hat{\boldsymbol{c}}\tilde{\boldsymbol{E}}\boldsymbol{\phi}\left(\tilde{\boldsymbol{E}}^\top\nabla\boldsymbol{C}\left(\boldsymbol{\pi}^I\left(\boldsymbol{s}\right)\right)\right) = \mathbb{0}_n$ if and only if $\nabla C(\boldsymbol{\pi}^I(\boldsymbol{s})) \in \mathrm{range}(\mathbb{1}_n)$. □

B.2 Proof of Theorem 3

Proof. At the equilibrium, we have $\omega_i - \frac{1}{n}\sum_{j=1}^n \omega_j = 0, \forall i$ and thus $\boldsymbol{\omega}^* = \hat{\omega}\mathbb{1}_n$ for some $\hat{\omega} \in \mathbb{R}$. The right side of (3.6b) equals to zero at the equilibrium gives $(\hat{\omega}\mathbb{1}_n) = -\hat{\boldsymbol{c}}\boldsymbol{E}\boldsymbol{\phi}\left(\boldsymbol{E}^\top\nabla\boldsymbol{C}(\boldsymbol{\pi}^I(\boldsymbol{s}))\right)$. Multiplying both sides by $\mathbb{1}_n^\top\hat{\boldsymbol{c}}^{-1}$ yields $(\hat{\omega})\mathbb{1}_n^\top\hat{\boldsymbol{c}}^{-1}\mathbb{1}_n = -\mathbb{1}_n^\top\boldsymbol{E}\boldsymbol{\phi}\left(\boldsymbol{E}^\top\nabla\boldsymbol{C}(\boldsymbol{\pi}^I(\boldsymbol{s}))\right)$, which equals to zero since $\mathbb{1}_n^\top\boldsymbol{E} = \mathbb{0}_n$ for a connected graph. This implies $\hat{\omega} = 0$ since $\mathbb{1}_n^T\hat{\boldsymbol{c}}^{-1}\mathbb{1}_n > 0$ for $\hat{\boldsymbol{c}} \succ 0$. Therefore, $\boldsymbol{\omega}^* = \mathbb{0}_n$.

Moreover, $\boldsymbol{\omega}^* = \mathbb{0}_n$ implies $\hat{\boldsymbol{c}}\tilde{\boldsymbol{E}}\boldsymbol{\phi}\left(\tilde{\boldsymbol{E}}^\top\nabla\boldsymbol{C}(\boldsymbol{\pi}^I\left(\boldsymbol{s}^*\right))\right) = \mathbb{0}_n$. By Lemma 8, $\nabla\boldsymbol{C}(\boldsymbol{\pi}^I(\boldsymbol{s})) \in \mathrm{range}(\mathbb{1}_n)$ and thus there exists a scalar γ such that $\nabla C_i(\pi_i^I\left(s_i^*\right)) = \gamma$ for all $i \in \mathcal{V}$. This implies $\nabla C_{\mathrm{o}}(c_i\pi_i^I\left(s_i^*\right)) = \gamma$ by Assumption 1. The strict convexity of $C_{\mathrm{o}}(\cdot)$ implies that $\nabla C_{\mathrm{o}}(\cdot)$ is a strictly increasing function, which guarantees the existence of $\nabla C_{\mathrm{o}}^{-1}(\cdot)$ that is also a strictly increasing function. Hence, $\pi_i^I(s_i^*) = \nabla C_{\mathrm{o}}^{-1}(\gamma)c_i^{-1}$ and compactly we have $\boldsymbol{\pi}^I\left(\boldsymbol{s}^*\right) = \nabla C_{\mathrm{o}}^{-1}(\gamma)\hat{\boldsymbol{c}}^{-1}\mathbb{1}_n$.

From $\boldsymbol{\pi}^P(\boldsymbol{\omega}^*) = \mathbb{0}_n$, we have $\boldsymbol{u}^* = \boldsymbol{\pi}^I\left(\boldsymbol{s}^*\right)$ and therefore $\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}) = \boldsymbol{p}_{\mathrm{m}} - \nabla C_{\mathrm{o}}^{-1}(\gamma)\hat{\boldsymbol{c}}^{-1}\mathbb{1}_n$. Since $\mathbb{1}_n^\top\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}) = 0$, we have $\nabla C_{\mathrm{o}}^{-1}(\gamma) = -\left(\sum_{i=1}^n p_{\mathrm{m},i}\right) / \left(\sum_{i=1}^n c_i^{-1}\right)$. The uniqueness of γ is guaranteed by the strict increasing property of function $\nabla C_{\mathrm{o}}^{-1}(\gamma)$. Similarly, the uniqueness of $\boldsymbol{s}^*$ satisfying $\boldsymbol{\pi}^I\left(\boldsymbol{s}^*\right) = \nabla C_{\mathrm{o}}^{-1}(\gamma)\hat{\boldsymbol{c}}^{-1}\mathbb{1}_n$ is guaranteed by the strictly increasing property of function $\pi_i^I(\cdot)$ for $i \in \mathcal{V}$.

□

Appendix C

APPENDIX FOR CHAPTER 5

C.1 Fundemental Lemma

For state feedback control, $\boldsymbol{\omega}(t) = \boldsymbol{x}(t)$ ($\boldsymbol{C} = \boldsymbol{I}_n$) and the system transition dynamics is reduced to $\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k)$. Expanding the dynamics for $k = 0, \cdots, T$ gives the input-state response over $[0, T-1]$ as

$$\begin{bmatrix} \boldsymbol{u}_{[0,T-1]} \\ \boldsymbol{x}_{[0,T-1]} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{Tm} & \mathbb{0}_{Tm\times n} \\ \mathcal{T}_{[0,T-1]} & \mathcal{O}_{[0,T-1]} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_{[0,T-1]} \\ \boldsymbol{x}(0) \end{bmatrix}, \tag{C.1}$$

where $\mathcal{T}_{[0,T-1]}$ and $\mathcal{O}_{[0,T-1]}$ are the Toeplitz and observability matrices of order T represented as [118]

$$\mathcal{T}_{[0,T-1]} := \begin{bmatrix} \boldsymbol{B} & \mathbb{0}_{n\times m} & \mathbb{0}_{n\times m} & \cdots & \mathbb{0}_{n\times m} \\ \boldsymbol{A}\boldsymbol{B} & \boldsymbol{B} & \mathbb{0}_{n\times m} & \cdots & \mathbb{0}_{n\times m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}^{T-2}\boldsymbol{B} & \boldsymbol{A}^{T-3}\boldsymbol{B} & \boldsymbol{A}^{T-4}\boldsymbol{B} & \cdots & \mathbb{0}_{n\times m} \end{bmatrix} \quad \mathcal{O}_{[0,T-1]} := \begin{bmatrix} \boldsymbol{I}_n \\ \boldsymbol{A} \\ \vdots \\ \boldsymbol{A}^{T-1} \end{bmatrix}.$$

On this basis, the Hankel Matrix $\mathcal{H}$ can be represented as

$$\begin{bmatrix} \boldsymbol{U}_{0,T,L-T+1} \\ \boldsymbol{X}_{0,T,L-T+1} \end{bmatrix} = \begin{bmatrix} I_{Tm} & \mathbb{0}_{Tm\times n} \\ \mathcal{T}_{[0,T-1]} & \mathcal{O}_{[0,T-1]} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_{0,T,L-T+1} \\ X_{0,L-T+1} \end{bmatrix}, \tag{C.2}$$

where $X_{0,L-T+1} := \begin{bmatrix} \boldsymbol{x}_d(0) & \boldsymbol{x}_d(1) & \ldots & \boldsymbol{x}_d(L-T) \end{bmatrix}$.

Now consider a trajectory $[\hat{\boldsymbol{u}}_{[0,T-1]}; \hat{\boldsymbol{x}}_{[0,T-1]}]$ starting from an initial state $\hat{\boldsymbol{x}}(0)$ and evolves with the sequence of actions $\hat{\boldsymbol{u}}_{[0,T-1]}$. If $\begin{bmatrix} \boldsymbol{U}_{0,T,L-T+1} \\ X_{0,L-T+1} \end{bmatrix}$ is full row rank, namely rank($\begin{bmatrix} \boldsymbol{U}_{0,T,L-T+1} \\ X_{0,L-T+1} \end{bmatrix}$) =

$n + Tm$, then there exists $\hat{\boldsymbol{g}} \in \mathbb{R}^{L-T+1}$ such that $\begin{bmatrix} \hat{\boldsymbol{u}}_{[0,T-1]} \\ \hat{\boldsymbol{x}}(0) \end{bmatrix} = \begin{bmatrix} \boldsymbol{U}_{0,T,L-T+1} \\ X_{0,L-T+1} \end{bmatrix} \hat{\boldsymbol{g}}$. By (C.1),

$$\begin{aligned} \begin{bmatrix} \hat{\boldsymbol{u}}_{[0,T-1]} \\ \hat{\boldsymbol{x}}_{[0,T-1]} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{I}_{Tm} & \mathbb{0}_{Tm\times n} \\ \mathcal{T}_{[0,T-1]} & \mathcal{O}_{[0,T-1]} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_{0,T,L-T+1} \\ X_{0,L-T+1} \end{bmatrix} \hat{\boldsymbol{g}} \\ &= \begin{bmatrix} \boldsymbol{U}_{0,T,L-T+1} \\ \boldsymbol{X}_{0,T,L-T+1} \end{bmatrix} \hat{\boldsymbol{g}}, \end{aligned} \tag{C.3}$$

where the second equation follows from the relation in (C.2). This complete the proof of Lemma (14).

C.2 Policy gradient algorithm

Algorithm 4: Policy Gradient with trajectory generation

Require: The length T of trajectory, the learning rate α, total number of episode I

Policy Gradient with Data generation:*Initialization* :Initial weights $\boldsymbol{\theta}$ for control network

for $episode = 1$ *to* I **do**

  Generate a batch of Q trajectories $[\boldsymbol{\tau}_1, \cdots, \boldsymbol{\tau}_Q]$ =`TrajectoryGen`$(\mathcal{H}, \boldsymbol{\theta}, \mathcal{D}, Q)$;

  Compute the gradient $\nabla J(\boldsymbol{\theta}) = \frac{1}{Q} \sum_{i=1}^{Q} c(\boldsymbol{\tau}_i) \sum_{t=1}^{T} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(\tilde{\boldsymbol{u}}_i(t)|\tilde{\boldsymbol{x}}_i(t))$;

  Update weights in the neural network by gradient descent: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla J(\boldsymbol{\theta})$

end

C.3 Transition dynamics with extended states

Expanding the transition dynamics in (6.1) from time 0 to T_0 gives

$$\boldsymbol{\omega}_{[0,T_0-1]} = \mathcal{O}_{[0,T_0-1]} \boldsymbol{x}(0) + \mathcal{T}_{[0,T_0-1]} \boldsymbol{u}_{[0,T_0-2]} \tag{C.4}$$

where $\mathcal{T}_{[0,T-1]}$ and $\mathcal{O}_{[0,T-1]}$ are the Toeplitz and observability matrices of order T_0 represented as

$$\mathcal{T}_{[0,T_0-1]} := \begin{bmatrix} \boldsymbol{CB} & \mathbb{0}_{d\times m} & \mathbb{0}_{d\times m} & \cdots & \mathbb{0}_{d\times m} \\ \boldsymbol{CAB} & \boldsymbol{CB} & \mathbb{0}_{d\times m} & \cdots & \mathbb{0}_{d\times m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{CA}^{T_0-2}\boldsymbol{B} & \boldsymbol{CA}^{T_0-3}\boldsymbol{B} & \boldsymbol{CA}^{T_0-4}\boldsymbol{B} & \cdots & \mathbb{0}_{d\times m} \end{bmatrix} \quad \mathcal{O}_{[0,T_0-1]} := \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \vdots \\ \boldsymbol{CA}^{T_0-1} \end{bmatrix}.$$

Since the system (6.1) is observable, $\mathcal{O}_{[0,T_0-1]}$ is full column rank. Thus,

$$\boldsymbol{x}(0) = \underbrace{\left(\mathcal{O}_{[0,T_0-1]}^\top \mathcal{O}_{[0,T_0-1]}\right)^{-1} \mathcal{O}_{[0,T_0-1]}^\top}_{\mathcal{O}_{[0,T_0-1]}^\dagger} \left(\boldsymbol{\omega}_{[0,T_0-1]} - \mathcal{T}_{[0,T_0-1]}\boldsymbol{u}_{[0,T_0-2]}\right) \tag{C.5}$$

Then pluging in the expression of $\boldsymbol{\omega}(T_0)$ yields

$$\begin{aligned} \boldsymbol{\omega}(T_0) &= \boldsymbol{CA}^{T_0}\boldsymbol{x}(0) + \boldsymbol{CA}^{T_0-1}\boldsymbol{Bu}(0) + \cdots + \boldsymbol{CBu}(T_0-1) \\ &= \boldsymbol{CA}^{T_0}\mathcal{O}_{[0,T_0-1]}^\dagger \left(\boldsymbol{\omega}_{[0,T_0-1]} - \mathcal{T}_{[0,T_0-1]}\boldsymbol{u}_{[0,T_0-2]}\right) + \mathcal{T}_{[T_0,T_0]}\boldsymbol{u}_{[0,T_0-2]} \\ &= \boldsymbol{CA}^{T_0}\mathcal{O}_{[0,T_0-1]}^\dagger \boldsymbol{\omega}_{[0,T_0-1]} + \left(\mathcal{T}_{[T_0,T_0]} - \boldsymbol{CA}^{T_0}\mathcal{O}_{[0,T_0-1]}^\dagger \mathcal{T}_{[0,T_0-1]}\right)\boldsymbol{u}_{[0,T_0-2]} \end{aligned} \tag{C.6}$$

where

$$\mathcal{T}_{[T_0,T_0]} := \begin{bmatrix} \boldsymbol{CA}^{T_0-1}\boldsymbol{B} & \boldsymbol{CA}^{T_0-2}\boldsymbol{B} & \boldsymbol{CA}^{T_0-3}\boldsymbol{B} & \cdots & \boldsymbol{CB} \end{bmatrix}.$$

Stacking the observations and the outputs together yields

$$\begin{bmatrix} \boldsymbol{\omega}(1) \\ \vdots \\ \boldsymbol{\omega}(T_0) \\ \boldsymbol{u}(1) \\ \vdots \\ \boldsymbol{u}(T_0-1) \end{bmatrix} = \tilde{\boldsymbol{A}} \begin{bmatrix} \boldsymbol{\omega}(0) \\ \vdots \\ \boldsymbol{\omega}(T_0-1) \\ \boldsymbol{u}(0) \\ \vdots \\ \boldsymbol{u}(T_0-2) \end{bmatrix} + \tilde{\boldsymbol{B}}\boldsymbol{u}(T_0-1), \tag{C.7}$$

where

$$\tilde{\boldsymbol{A}} = \left(\begin{array}{cccccc|cccccc} \boldsymbol{0} & \boldsymbol{I}_d & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & & & & & & \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_d & \boldsymbol{0} & \cdots & \boldsymbol{0} & & & & & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & & & & & & \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{I}_d & & & & & & \\ \hline \multicolumn{6}{c|}{\boldsymbol{C}\boldsymbol{A}^{T_0}\mathcal{O}^{\dagger}_{[0,T_0-1]}} & \multicolumn{6}{c}{\mathcal{T}_{[T_0,T_0]} - \boldsymbol{C}\boldsymbol{A}^{T_0}\mathcal{O}^{\dagger}_{[0,T_0-1]}\mathcal{T}_{[0,T_0-1]}} \\ \hline & & & & & & \boldsymbol{0} & \boldsymbol{I}_m & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ & & & & & & \boldsymbol{0} & 0 & \boldsymbol{I}_m & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ & & & & & & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ & & & & & & \boldsymbol{0} & 0 & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{I}_m \\ \hline & & & & & & & & & & & \end{array}\right), \tilde{\boldsymbol{B}} = \left(\begin{array}{c} \boldsymbol{0} \\ \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \\ \hline \boldsymbol{0} \\ \hline \boldsymbol{0} \\ \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \\ \hline \boldsymbol{I}_m \end{array}\right). \tag{C.8}$$

C.4 Policy Gradient for Extended States

The basic policy gradient algorithm is [53]

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{\tau}\sim p_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}}[c(\boldsymbol{\tau})] &= \nabla_{\boldsymbol{\theta}} \int \pi_{\boldsymbol{\theta}}(\boldsymbol{\tau}) c(\boldsymbol{\tau}) d\tau \\ &= \int \nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}}(\boldsymbol{\tau}) c(\boldsymbol{\tau}) d\tau \\ &= \int \pi_{\boldsymbol{\theta}}(\boldsymbol{\tau}) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(\boldsymbol{\tau}) c(\boldsymbol{\tau}) d\tau \\ &= \mathbb{E}_{\boldsymbol{\tau}\sim p_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}}[c(\boldsymbol{\tau}) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(\boldsymbol{\tau})]. \end{aligned} \tag{C.9}$$

The probability of a trajectory with length T is

$$\pi_{\theta}(\boldsymbol{\tau}) = p\left(\mathcal{X}\left(T_0\right)\right) \prod_{t=T_0}^{T-1} p_{\pi_{\boldsymbol{\theta}}}\left(\boldsymbol{u}(t) | \mathcal{X}(t)\right) p\left(\mathcal{X}(t+1) | \mathcal{X}(t), \boldsymbol{u}(t)\right). \tag{C.10}$$

Expanding the terms in (C.10) and canceling the transition probability independent of $\boldsymbol{\theta}$, we have

$$\nabla_{\boldsymbol{\theta}} \log \pi_{\theta}(\boldsymbol{\tau}) = \sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\left(\boldsymbol{u}(k) \mid \boldsymbol{\omega}(k)\right), \tag{C.11}$$

and therefore,

$$\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\pi_\theta}[c(\boldsymbol{\tau})] = \mathbb{E}_{\boldsymbol{\tau}\sim p_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{\tau})}\left[c(\boldsymbol{\tau})\sum_{k=0}^{K-1}\nabla_{\boldsymbol{\theta}}\log p_{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\left(\boldsymbol{u}(k)\mid\boldsymbol{\omega}(k)\right)\right] \tag{C.12}$$

Hence, the policy gradient algorithm still holds for the output-feedback case.

C.5 Proof of Theorem 9

To prove Theorem 9, we need to make use of the rank condition of $\boldsymbol{G_\theta}$ and $\boldsymbol{R}$ in (6.16). We first show in Lemma 16 about the null space and rank condition induced from the condition rank$(\mathcal{H}) = n + Tm$.

Lemma 16. *If the observability matrix $\mathcal{O}_{[0,T_0-1]}$ is full column rank, then the null space $\mathcal{N}(\boldsymbol{G_\theta})$ is the same as $\mathcal{N}(\mathcal{H})$. Moreover, if rank$(\mathcal{H}) = n + Tm$, then rank$(\boldsymbol{G_\theta}) = n + Tm$.*

Proof. We first prove that the null space $\mathcal{N}(\boldsymbol{G_\theta})$ is the same as $\mathcal{N}(\mathcal{H})$ from *(i)* and *(ii)* :

(i) For all $\boldsymbol{q} \in \mathcal{N}(\mathcal{H})$, we have $[\mathcal{H}_y]\boldsymbol{q} = \mathbb{0}_{Tn}$ and $[\mathcal{H}_u]\boldsymbol{q} = \mathbb{0}_{Tm}$. Plugging in the expression of $\boldsymbol{G_\theta}$ yields $\boldsymbol{G_\theta q} = \mathbb{0}_{Tm+n}$. Namely, $\boldsymbol{q} \in \mathcal{N}(\boldsymbol{G_\theta})$.

(ii) For all $\boldsymbol{v} \in \mathcal{N}(\boldsymbol{G_\theta})$, we have

$$\begin{bmatrix} \mathcal{H}_u^{T_0-1:T-1} - (\boldsymbol{I}_{T-T_0}\otimes\theta)\,\mathcal{H}_y^{T_0-1:T-1} \\ \mathcal{H}_y^{0:T_0-1} \\ \mathcal{H}_u^{0:T_0-2} \end{bmatrix}\boldsymbol{v} = \mathbb{0}_{Tm+T_0d}\ , \tag{C.13}$$

which gives

$$\begin{aligned} &\mathcal{H}_u^k\boldsymbol{v} = \theta\mathcal{H}_y^k\boldsymbol{v} \text{ for } k = T_0,\cdots,T-1 \\ &\mathcal{H}_y^{0:T_0-1}\boldsymbol{v} = \mathbb{0}_{T_0d} \\ &\mathcal{H}_u^{0:T_0-2}\boldsymbol{v} = \mathbb{0}_{T_0m}\ . \end{aligned} \tag{C.14}$$

From (C.6), we have ,

$$\mathcal{H}_y^k = \boldsymbol{C}\boldsymbol{A}^{T_0}\mathcal{O}^\dagger_{[0,T_0-1]}\mathcal{H}_y^{k-T_0:k-1} + \left(\mathcal{T}_{[T_0,T_0]} - \boldsymbol{C}\boldsymbol{A}^{T_0}\mathcal{O}^\dagger_{[0,T_0-1]}\mathcal{T}_{[0,T_0-1]}\right)\mathcal{H}_u^{k-T_0:k-2} \tag{C.15}$$

for $k = T_0, \cdots, T-1$.

Plugging (C.14) in (C.15) induces $\mathcal{H}_y^k \boldsymbol{v} = \mathbb{0}_d$ and $\mathcal{H}_u^k \boldsymbol{v} = \mathbb{0}_m$ for $k = 0, \cdots, T-1$. Hence, $\mathcal{H}\boldsymbol{v} = \mathbb{0}_{Tm+Td}$. Namely, $\boldsymbol{v} \in \mathcal{N}(\mathcal{H})$.

Next, we prove the rank condition. Note that $\mathcal{H} \in \mathbb{R}^{(Tm+Tn)\times(L-T+1)}$. If rank$(\mathcal{H}) = n + Tm$, then the rank of Null space is rank$(\mathcal{N}(\mathcal{H})) = (L-T+1) - (n+Tm)$. Since $\mathcal{N}(\boldsymbol{G_\theta})$ is the same as $\mathcal{N}(\mathcal{H})$, then rank$(\mathcal{N}(\boldsymbol{G_\theta})) = (L-T+1) - (n+Tm)$. It follows directly that rank$(\boldsymbol{G_\theta}) =$ $(L-T+1) -$ rank$(\mathcal{N}(\boldsymbol{G_\theta})) = n + Tm$. □

Then, we are ready to prove Theorem 9 as follows.

Proof. We first prove the existence of the solution in (6.16). From (C.4), we have

$$\mathcal{X}(T_0-1) := \begin{bmatrix} \boldsymbol{u}_{[0,T_0-2]} \\ \boldsymbol{\omega}_{[0,T_0-1]} \end{bmatrix} = \begin{bmatrix} I_{(T_0-1)m} & \mathbb{0}_{(T_0-1)m\times n} \\ \mathcal{T}_{[0,T_0-1]} & \mathcal{O}_{[0,T_0-1]} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_{[0,T_0-2]} \\ \boldsymbol{x}(0) \end{bmatrix}$$

The number of element in the vector $[\boldsymbol{u}_{[0,T_0-2]}^\top, \boldsymbol{x}^\top(0)]^\top$ is $n + m(T_0-1)$. Hence, the rank of $\mathcal{X}(T_0-1)$ is as most $n+m(T_0-1)$. Then the right side of (6.16) has the rank as most $n+m(T_0-1)+m(T-T_0+1) = n+Tm$. By Lemma 16, rank$(\mathcal{H}) = n+Tm$ yields rank$(\boldsymbol{G_\theta}) = n+Tm$. Hence, there exists at least one solution such that (6.10) holds.

Next, we show the uniqueness of the generated trajectory. Suppose there exists $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ are both solution of (6.10) and $\mathcal{H}\boldsymbol{g}_1 \neq \mathcal{H}\boldsymbol{g}_2$. Since $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ are both solution of (6.10), then $\boldsymbol{G_\theta}\boldsymbol{g}_1 = \boldsymbol{G_\theta}\boldsymbol{g}_1$ and thus $(\boldsymbol{g}_1 - \boldsymbol{g}_2) \in \mathcal{N}(\boldsymbol{G_\theta})$. On the other hand, $\mathcal{H}\boldsymbol{g}_1 \neq \mathcal{H}\boldsymbol{g}_2$ yields $\mathcal{H}(\boldsymbol{g}_1 - \boldsymbol{g}_2) \neq 0$ and thus $(\boldsymbol{g}_1 - \boldsymbol{g}_2) \notin \mathcal{N}(\mathcal{H})$. This contradicts that $\mathcal{N}(\boldsymbol{G_\theta})$ is the same as $\mathcal{N}(\mathcal{H})$ proved in Lemma 16. Hence, $\mathcal{H}\boldsymbol{g}_1 = \mathcal{H}\boldsymbol{g}_2$, namely, the generated trajectories are identical. □

C.6 Trajectory generation algorithm for output-feedback control

Algorithm 5: Trajectory generation for output-feedback control

Data collection: Collect historic measurement of the system and stack each T-length input-output trajectory as Hankel matrix $\mathcal{H}$ shown in (6.5) until rank$(\mathcal{H}) = n + Tm$

Data generation:*Input* :Hankel matrix $\mathcal{H}$, weights $\boldsymbol{\theta}$ and the distribution $\mathcal{D}$ for the control policy, the batchsize Q for the generated trajectories, the set $\mathcal{S}_{\mathcal{X}}$ of historic initial extended state $\mathcal{X}(T_0 - 1)$

Function `TrajectoryGen(`$\mathcal{H}, \boldsymbol{\theta}, \mathcal{D}, Q, \mathcal{S}_{\mathcal{X}}$`)`**:**

  Plug in $\boldsymbol{\theta}$ to compute $\boldsymbol{G}_{\boldsymbol{\theta}}$ in (6.16).

  Conduct eigenvalue decomposition of $\left(\boldsymbol{G}_{\boldsymbol{\theta}}\boldsymbol{G}_{\boldsymbol{\theta}}^{\top}\right)$ to obtain $\boldsymbol{P}_{\boldsymbol{\theta}} := \begin{bmatrix}\boldsymbol{p}_1 & \boldsymbol{p}_2 & \cdots & \boldsymbol{p}_s\end{bmatrix}$ and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_s)$ with λ_i being nonzero eigenvalues and $\boldsymbol{p}_i$ being orthonormal eigenvectors.

  for $i = 1$ *to* Q **do**

    Sample $\mathcal{X}(T_0 - 1)$ from $\mathcal{S}_{\mathcal{X}}$. Sample $\{\boldsymbol{w}_i(T_0 - 1), \cdots, \boldsymbol{w}_i(T - 1)\}$ from distribution $\mathcal{D}$.

    Compute the coefficient $\boldsymbol{g}_i^* = \boldsymbol{G}_{\boldsymbol{\theta}}^{\top}\boldsymbol{P}_{\boldsymbol{\theta}}\boldsymbol{\Lambda}^{-1}\boldsymbol{P}_{\boldsymbol{\theta}}^{\top}\left[\boldsymbol{w}_i(T_0 - 1)^{\top} \cdots \boldsymbol{w}_i(T - 1)^{\top}\mathcal{X}(T_0 - 1)^{\top}\right]^{\top}$.

    Generate the i-th trajectory $\boldsymbol{\tau}_i :=$

$$\left[\tilde{\boldsymbol{u}}_i(T_0 - 1)^{\top} \cdots \tilde{\boldsymbol{u}}_i(T - 1)^{\top}\tilde{\boldsymbol{\omega}}_i(T_0 - 1)^{\top} \cdots \tilde{\boldsymbol{\omega}}_i(T - 1)^{\top}\right]^{\top} = \begin{bmatrix}\mathcal{H}_u^{T_0-1:T-1} \\ \mathcal{H}_y^{T_0-1:T-1}\end{bmatrix} \boldsymbol{g}_i^*.$$

  end

  return $[\boldsymbol{\tau}_1, \cdots, \boldsymbol{\tau}_Q]$