Reinforcement Learning for Optimal Primary Frequency Control: A Lyapunov Approach

Wenqi Cui, Yan Jiang, and Baosen Zhang

***Abstract*—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.

***Index Terms*—Power system dynamics, Primary frequency control, Nonlinear Systems, Reinforcement learning**

I. 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 [1]. 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) [2]. Inverter-connected resources can mimic this behavior by changing their active power setpoints subject to frequency deviations [3], [4]. However, as for the common performance metrics adopted in practice, including frequency deviations and control costs [5], [6], linear controllers are not optimal [7]. 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 [8], [9]. 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 [5], [6]. 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 [4]. To obtain better performances, model predictive control has also been used [7], [9], 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 [10]–[12]. 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 [10], [13]. 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

The authors are with the Department of Electrical and Computer Engineering, University of Washington, Seattle, WA, 98195 e-mail:{wenqicui,jiangyan, zhangbao}@uw.edu

The authors are supported in part by the National Science Foundation grant ECCS-1930605, ECCS-1942326 and the Washington Clean Energy Institute.

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 [14], which is generally not true during transients. Lyapunov functions have been used as constraints [15], but learning was not considered and controllers was manually tuned.

This paper 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.

In summary, the main contributions of the paper are:

1) A Lyapunov function is integrated in the structural properties of controllers, guaranteeing local asymptotic stability over a large set of states. Namely, the controllers need to be monotonic functions crossing the origin.
2) The controller is parameterized with a stacked-ReLU neural network and a RNN-based RL framework is proposed to efficiently train the controllers.

The remaining of this paper is organized as follows. Section II introduces the system model and the optimal control problem. Section III provides the main theorems governing the structure of a stabilizing controller and illustrates how it can be achieved via neural networks. Section IV shows how they can be trained efficiently. Section V shows the simulation results. Section VI concludes the paper.

II. Model and Problem Formulation

A. Power System Model

Consider a n-bus power system that can be modelled as a connected graph $(\mathcal{V}, \mathcal{E})$. Specifically, buses are indexed by $i, j \in \mathcal{V} := [n] := \{1, \dots, n\}$ 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 value $\boldsymbol{\omega} := (\omega_i, i \in [n]) \in \mathbb{R}^n$.[1]

In this paper, 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 and storages).

We assume the bus voltage magnitudes are 1 per unit and the reactive power flows and injections 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 [16]. Let $p_{\mathrm{m},i}$, $p_{\mathrm{e},i}$ and $p_{\mathrm{l},i}$ be the mechanical power, electrical power, and load at bus i, respectively. Denote M_i and L_i as the inertia constant and load damping coefficient at bus i. We assume coherency between the internal (rotor) angle and terminal (bus) voltage phase angles of the synchronous generators [5], [17]–[19] (numerical validation are given in Appendix D). Then, the frequency dynamics[2] is given by the swing equation [2]:

$$\dot{\theta}_i = \omega_i \,, \tag{1a}$$

$$M_i \dot{\omega}_i = p_{\mathrm{m},i} - p_{\mathrm{l},i} - p_{\mathrm{e},i} - L_i \omega_i - p_{\mathrm{re},i} \,. \tag{1b}$$

Here, $p_{\mathrm{re},i}$ is the active power injection from inverter-connected resources at bus i, which follows the setpoint given by the control law $u_i(\omega_{\mathrm{pll},i})$, i.e., $p_{\mathrm{re},i} = u_i(\omega_{\mathrm{pll},i})$, where $\omega_{\mathrm{pll},i}$ is the frequency deviation of bus i read from the phase-locked-loop (PLL). The diagram of the frequency control loop for the system (1) is shown in Fig. 1 [2], [20]. All of our simulations use the 6^{th}-order generator model with turbine-governing system and we use dynamic model for inverter-connected resources (blue blocks in Fig. 1). The speed droop response with coefficient $\frac{1}{R_i}$ for synchronous generator at bus i is implemented through turbine-governing systems, see [2, Chapter 11] for details.

As in most existing literature [21]–[24], we use the classical 2^{nd}-order model for synchronous generators in theoretical analysis. Under the classical 2^{nd}-order model, the electrical power is $p_{\mathrm{e},i} = \sum_{j=1}^{n} B_{ij} \sin(\theta_i - \theta_j)$ and the mechanical power is $p_{\mathrm{m},i} = p_{\mathrm{g},i} - \frac{1}{R_i}\omega_i$ with $p_{\mathrm{g},i}$ being the active power setpoint of the generator. Let $p_i := p_{\mathrm{g},i} - p_{\mathrm{l},i}$ represent the net power injection of bus i at the generator setpoint. Let $D_i := \frac{1}{R_i} + L_i$ be the combined frequency response coefficient from synchronous generators and load. Considering the much faster time response of PLLs and inverters [25], we assume that the measured frequency deviation $\omega_{\mathrm{pll},i}$ accurately approximate ω_i and thus $p_{\mathrm{re},i} = u_i(\omega_i)$. Then, the system dynamics in (1b) becomes

$$M_i \dot{\omega}_i = p_i - D_i \omega_i - u_i(\omega_i) - \sum_{j=1}^{n} B_{ij} \sin(\theta_i - \theta_j), \tag{2}$$

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

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

[2] The phase angle dynamics are utilized to derive the structure of stabilizing controller. We do not make any changes to the rotor angle control on synchronous generators.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2022.3176525, IEEE Transactions on Power Systems

combined frequency response coefficients from synchronous generators and frequency sensitive load, $\boldsymbol{p} := (p_i, i \in [n]) \in \mathbb{R}^n$ 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 $\boldsymbol{u}(\boldsymbol{\omega}) := (u_i(\omega_i), i \in [n]) \in \mathbb{R}^n$.

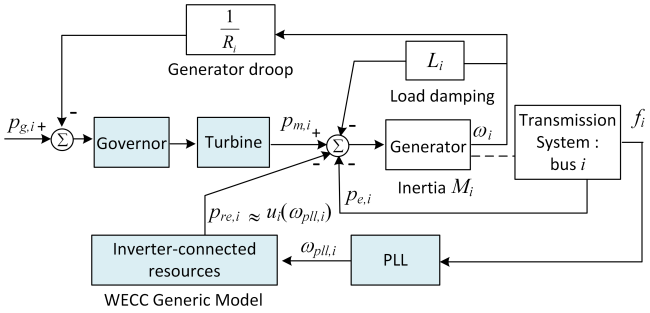


Fig. 1. Block diagram of frequency control loop [2]. The blue blocks constitute the dynamics of the system. High-order models are used in simulations and simplified models are used for analysis.

B. Optimization Problem Formulation

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 [6], [7], [26], [27]. 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. The cost on control actions is a Lipschitz-continuous function written as $C_i(u_i)$ for bus $i = 1, \cdots, n$. For example, for a battery, we can set C to reflect its operating (i.e., degradation and energy) cost as in [28], [29]. The optimization problem is:

$$\min_{\boldsymbol{u}} \sum_{i=1}^{n} \left(\|\omega_i\|_\infty + \gamma C_i(u_i)\right) \tag{3a}$$

$$\text{s.t. } \dot{\theta}_i = \omega_i \tag{3b}$$

$$M_i\dot{\omega}_i = p_i - D_i\omega_i - u_i(\omega_i) - \sum_{j=1}^{n} B_{ij}\sin(\theta_i - \theta_j) \tag{3c}$$

$$\underline{u}_i \le u_i(\omega_i) \le \overline{u}_i \tag{3d}$$

$$u_i(\omega_i) \text{ is stabilizing.} \tag{3e}$$

Here, γ in (3a) 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 [23]. 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 (3d). 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 (1) as a hard constraint in (3e).

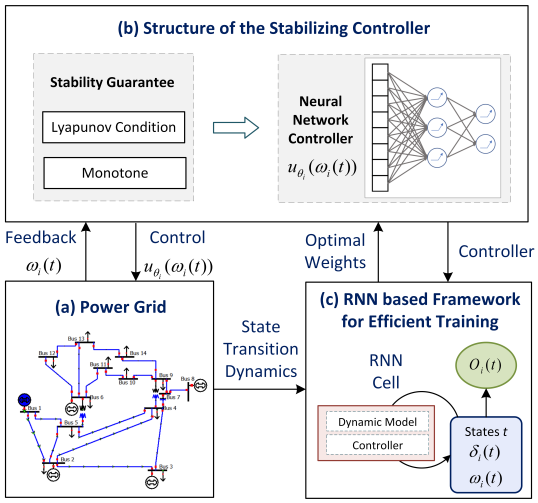


Fig. 2. Reinforcement learning for the frequency control problem

C. Reinforcement Learning for Optimal Frequency Control

In (3), 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 (3a).

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 (1) to derive the structure of the stabilizing controller based on Lyapunov stability theory. As illustrated in Fig. 2(b) and discussed in Section III, stability can be guaranteed by enforcing a structure on the controllers $u_{\varphi_i}(\omega_i)$'s.

To use RL, we need to discretize the system dynamics in (1). The weights φ_i's impact system behaviors across all of the time steps, which makes direct back propagation inefficient. Thus, we use the state transition dynamics to create a RNN framework to increase training efficiency, as illustrated in Fig. 2(c). Details are elaborated in Section IV.

III. Structural Properties of the Controller

To constrain the search space in (3) 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 (3a).

A. Uniqueness of the Equilibrium

Since the frequency dynamics of the system in (1b) depends only on the phase angle differences, to characterize the equilibrium of the dynamics (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 [16], [30]. Then, the system dynamics in (1) can be written as

$$\dot{\delta}_i = \omega_i - \frac{1}{n}\sum_{j=1}^{n}\omega_j\,, \tag{4a}$$

$$M_i\dot{\omega}_i = p_i - D_i\omega_i - u_i(\omega_i) - \sum_{j=1}^{n} B_{ij}\sin(\delta_i - \delta_j)\,. \tag{4b}$$

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 at the equilibrium 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*

$$0 = p_i - D_i\omega^* - u_i(\omega^*) - \sum_{j=1}^{n} B_{ij}\sin(\delta_i^* - \delta_j^*)\,, \tag{5a}$$

$$\sum_{i=1}^{n} p_i = \sum_{i=1}^{n} u_i(\omega^*) + \omega^*\sum_{i=1}^{n} D_i\,, \tag{5b}$$

if the power flow equations (5a) *are feasible, where* $\mathbf{1}$ *is a vector of all* 1*'s with an appropriate dimension.*

Proof. First of all, in steady state, (4) yields

$$0 = \omega_i^* - \frac{1}{n}\sum_{j=1}^{n}\omega_j^*\,, \tag{6a}$$

$$0 = p_i - D_i\omega_i^* - u_i(\omega_i^*) - \sum_{j=1}^{n} B_{ij}\sin(\delta_i^* - \delta_j^*)\,. \tag{6b}$$

Clearly, (6a) implies that the frequency deviation at each bus synchronizes to the same solution that $\omega_i^* = \omega^*$, and we have the desired equations in (5a). 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 (5a), we get (5b).

Next, we show the uniqueness of ω^* by contradiction. Suppose that both ω^* and $\hat{\omega}$ satisfy (5b), where $\omega^* \neq \hat{\omega}$. Then,

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

which yields

$$\sum_{i=1}^{n}\frac{u_i(\omega^*) - u_i(\hat{\omega})}{\omega^* - \hat{\omega}} = -\sum_{i=1}^{n} D_i < 0\,. \tag{7}$$

However, if $u_i(\omega_i)$ is monotonically increasing, the left hand side of the equality in (7) must be nonnegative, which is a contradiction. The uniqueness of $\boldsymbol{\delta}^*$ follows from the same argument as in [31, 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 [16], [30].

B. Lyapunov Stability Analysis

In this subsection, we further show that the equilibrium point $(\boldsymbol{\delta}^*, \boldsymbol{\omega}^*)$ described by (5) 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* (5) *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 [32]. 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 III-C) and the rest of this section outlines the proof of Theorem 1.

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 [32]. Therefore, we prove Theorem 1 by constructing a qualified Lyapunov function and showing that such a constant c exist. Inspired by [30], we consider the following Lyapunov function candidate:

$$V(\boldsymbol{\delta}, \boldsymbol{\omega}) = \frac{1}{2}\sum_{i=1}^{n} M_i(\omega_i - \omega^*)^2 + W_{\mathrm{p}}(\boldsymbol{\delta}) + \epsilon W_{\mathrm{c}}(\boldsymbol{\delta}, \boldsymbol{\omega}) \tag{8}$$

[3]These are sometimes called extended class κ functions

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2022.3176525, IEEE Transactions on Power Systems

with

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

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

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 [30], [33]. 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 $\dot{V}(\boldsymbol{\delta},\boldsymbol{\omega}) \le -cV(\boldsymbol{\delta},\boldsymbol{\omega})$.

Lemma 2 (Bounds on Lyapunov function). *$\forall(\boldsymbol{\delta},\boldsymbol{\omega})\in\mathcal{D}$, the Lyapunov function $V(\boldsymbol{\delta},\boldsymbol{\omega})$ in* (8) *satisfies*

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

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

The proof is given in Appendix A. 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* (8) *is given by*

$$\dot{V}(\boldsymbol{\delta},\boldsymbol{\omega}) = -\begin{bmatrix}\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta})-\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*)\\ \boldsymbol{\omega}-\boldsymbol{\omega}^*\end{bmatrix}^T \boldsymbol{Q}(\boldsymbol{\delta}) \begin{bmatrix}\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta})-\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*)\\ \boldsymbol{\omega}-\boldsymbol{\omega}^*\end{bmatrix} - \left[\boldsymbol{\omega}-\boldsymbol{\omega}^*+\epsilon\left(\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta})-\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*)\right)\right]^T\left(\boldsymbol{u}(\boldsymbol{\omega})-\boldsymbol{u}(\boldsymbol{\omega}^*)\right) \tag{9}$$

with

$$\boldsymbol{Q}(\boldsymbol{\delta}) := \begin{bmatrix}\epsilon\boldsymbol{I} & \frac{\epsilon}{2}\boldsymbol{D}\\ \frac{\epsilon}{2}\boldsymbol{D} & \boldsymbol{D}-\frac{\epsilon}{2}(\boldsymbol{H}(\boldsymbol{\delta})\boldsymbol{M}+\boldsymbol{M}\boldsymbol{H}(\boldsymbol{\delta}))\end{bmatrix}, \tag{10}$$

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

$$H_{ij} := \begin{cases} -B_{ij}\cos(\delta_{ij}) & \text{if } i\neq j\\ \sum\limits_{j'=1,j'\neq i}^{n} B_{ij'}\cos(\delta_{ij'}) & \text{if } i=j\end{cases}, \qquad \forall i,j\in[n]\,. \tag{11}$$

The proof is given in Appendix B. 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 $c > 0$ such that $\dot{V}(\boldsymbol{\delta},\boldsymbol{\omega}) \le -cV(\boldsymbol{\delta},\boldsymbol{\omega})$.*

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,

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

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 (9) as follows:

$$\begin{aligned}&\dot{V}(\boldsymbol{\delta},\boldsymbol{\omega}) \qquad\qquad (12)\\ &\le -\begin{bmatrix}\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta})-\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*)\\ \boldsymbol{\omega}-\boldsymbol{\omega}^*\end{bmatrix}^T\boldsymbol{Q}(\boldsymbol{\delta})\begin{bmatrix}\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta})-\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*)\\ \boldsymbol{\omega}-\boldsymbol{\omega}^*\end{bmatrix}\\ &\overset{(a)}{\le} -\lambda_{\min}(\boldsymbol{Q}(\boldsymbol{\delta}))\left(\|\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta})-\boldsymbol{p}_{\mathrm{e}}(\boldsymbol{\delta}^*)\|_2^2+\|\boldsymbol{\omega}-\boldsymbol{\omega}^*\|_2^2\right)\\ &\overset{(b)}{\le} -\lambda_{\min}(\boldsymbol{Q}(\boldsymbol{\delta}))\left(\gamma_1\|\boldsymbol{\delta}-\boldsymbol{\delta}^*\|_2^2+\|\boldsymbol{\omega}-\boldsymbol{\omega}^*\|_2^2\right)\\ &\le -\lambda_{\min}(\boldsymbol{Q}(\boldsymbol{\delta}))\min(1,\gamma_1)\left(\|\boldsymbol{\delta}-\boldsymbol{\delta}^*\|_2^2+\|\boldsymbol{\omega}-\boldsymbol{\omega}^*\|_2^2\right)\\ &\overset{(c)}{\le} -\lambda_{\min}(\boldsymbol{Q}(\boldsymbol{\delta}))\min(1,\gamma_1)\frac{1}{\alpha_2}V(\boldsymbol{\delta},\boldsymbol{\omega})\\ &\le -cV(\boldsymbol{\delta},\boldsymbol{\omega}) \qquad\qquad (13)\end{aligned}$$

with

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

where (a) is given by the Rayleigh-Ritz theorem, (b) is by [31, Lemma 4] with $\gamma_1 := \min_{\tilde{\delta}\in\Theta}\lambda_2(\boldsymbol{H}(\tilde{\boldsymbol{\delta}}))^2$, and (c) follows from Lemma 2. □

C. Design of Neural Network Controllers

In this paper, 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., $\sum_{i=1}^{n} p_i = 0$. 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;

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2022.3176525, IEEE Transactions on Power Systems

2) $u_{\varphi_i}(\omega_i) = 0$ for $\omega_i = 0$;
3) $\underline{u}_i \le u_{\varphi_i}(\omega_i) \le \overline{u}_i$ (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 $\omega_i \le 0$; $f_i^-(\omega_i)$ is monotonic increasing for $\omega_i < 0$ and zero when $\omega_i \ge 0$. 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* (14) *is monotonic increasing for $\omega_i > 0$ and zero when $\omega_i \le 0$.*

$$f_i^+(\omega_i) = \boldsymbol{q}_i \sigma(\mathbf{1}\omega_i + \boldsymbol{b}_i) \tag{14a}$$

$$\textit{where} \sum_{j=1}^{l} q_i^j \ge 0, \quad \forall l = 1, 2, \cdots, m \tag{14b}$$

$$b_i^1 = 0, b_i^l \le b_i^{(l-1)}, \quad \forall l = 2, 3, \cdots, m \tag{14c}$$

The stacked ReLU function constructed by (15) *is monotonic increasing for $\omega_i < 0$ and zero when $\omega_i \ge 0$.*

$$f_i^-(\omega_i) = \boldsymbol{z}_i \sigma(-\mathbf{1}\omega_i + \boldsymbol{c}_i) \tag{15a}$$

$$\textit{where} \sum_{j=1}^{l} z_i^j \le 0, \quad \forall l = 1, 2, \cdots, m \tag{15b}$$

$$c_i^1 = 0, c_i^l \le c_i^{(l-1)}, \quad \forall l = 2, 3, \cdots, m \tag{15c}$$

Proof. Note that the ReLU function $\sigma(x)$ is linear with x when activated ($x > 0$) and equals to zero when deactivated ($x \le 0$), 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. 3. Since $b_i^1 = 0$ and $b_i^l \le b_i^{(l-1)}, \forall 1 \le l \le m$, $g_i^l(\omega_i)$ 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., $\sum_{j=1}^{l} q_i^l \ge 0, \forall 1 \le l \le m$. Similarly, $f_i^-(\omega_i)$ also construct by ReLU function activated for negative w_i in sequence corresponding to c_i^l for $l = 1, \cdots, m$. $\sum_{j=1}^{l} z_i^j \le 0$ 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 (14) and (15), 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, (14b) is satisfied if $q_i^1 = \hat{q}_i^1$, $q_i^l = \hat{q}_i^l - \hat{q}_i^{(l-1)}$ for $l = 2, \cdots, m$. (14c) 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 $\hat{\boldsymbol{z}}_i = \begin{bmatrix}\hat{z}_i^1 & \cdots & \hat{z}_i^m\end{bmatrix} \ge 0$ and $\hat{\boldsymbol{c}}_i = \begin{bmatrix}\hat{c}_i^1 & \cdots & \hat{c}_i^m\end{bmatrix}^\intercal \ge 0$. Then, (15b) is satisfied if $z_i^1 = -\hat{z}_i^1$, $z_i^l = -\hat{z}_i^l + \hat{z}_i^{(l-1)}$ for $l = 2, \cdots, m$. (15c) 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 (14) and (15).[4]

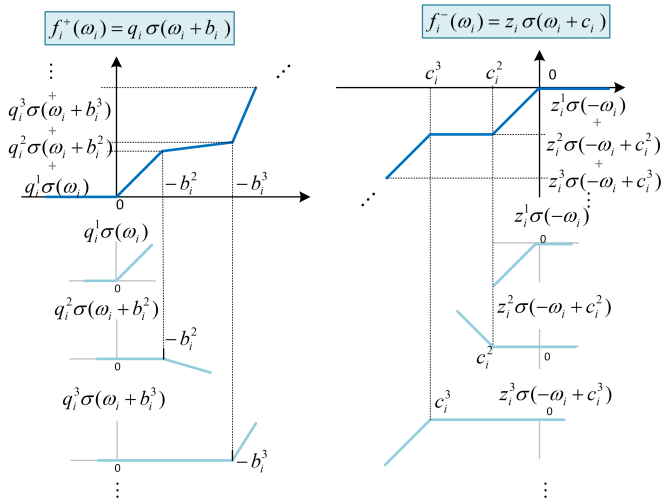


Fig. 3. Stacked ReLU neural network to formulate a monotonic increasing function through the origin

The next Theorem states the converse of Lemma 5, that is, the constructions in (14) and (15) suffice to approximate all functions of interest.

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 $\mathbb{R}$. For any $\epsilon > 0$, there exists a function $f(x) = f^+(x) + f^-(x)$ constructed by* (14) *and* (15) *such that $|r(x) - f(x)| < \epsilon$ when $x \in \mathbb{X}$.*

The proof is given in Appendix C. 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 paper 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 $i = 1, \cdots, N$:*

$$\begin{aligned} u_i(\omega_i) =& \overline{u}_i - \sigma(\overline{u}_i - f_i^+(\omega_i) - f_i^-(\omega_i)) \\ &+ \sigma(\underline{u}_i - f_i^+(\omega_i) - f_i^-(\omega_i)) \end{aligned} \tag{16}$$

The proof of this lemma is by inspection.

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2022.3176525, IEEE Transactions on Power Systems

IV. Learning Control Policies Using RNNs

The structure of the controllers are decided by the constructions in (14), (15) and (16). In this section we develop a RNN based RL algorithm to learn their weights and biases.

A. Discretize Time System

To learn the controller and simulate the trajectories of the system, we discretize the dynamics (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 The optimization problem is

$$\min_{\boldsymbol{\varphi}} \quad \sum_{i=1}^{n} \left(||\boldsymbol{\omega}_i||_\infty + \gamma C_i(\boldsymbol{u_{\varphi_i}})\right) \tag{17a}$$

$$\text{s.t. } \theta_i(k) = \theta_i(k-1) + \omega_i(k-1)\Delta t \tag{17b}$$

$$\omega_i(k) = -\frac{\Delta t}{M_i}\sum_{j=1}^{|\mathcal{B}|} B_{ij}\sin(\theta_{ij}(k-1)) + \frac{\Delta t}{M_i}p_{m,i} + \left(1 - \frac{D_i\Delta t}{M_i}\right)\omega_i(k-1) - \frac{\Delta t}{M_i}u_{\varphi_i}(\omega_i(k-1)) \tag{17c}$$

$$\underline{u}_i \le u_{\varphi_i}(\omega_i(k)) \le \overline{u}_i \tag{17d}$$

$$\omega_i(k)u_{\varphi_i}(\omega_i(k)) \ge 0 \tag{17e}$$

$$u_{\varphi_i}(\cdot) \text{ is increasing} \tag{17f}$$

and all equations hold for $i = 1, \ldots, n$. The constraints (17e) and (17f) guarantee exponentially stability.

Note that the optimization variable $\boldsymbol{\varphi}$ exists in all the time steps in (17). 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 (17b) and (17c). This way, the gradient of the optimization objective with respect to $\boldsymbol{\varphi}$ can be computed efficiently through back-propagation.

B. 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. 4

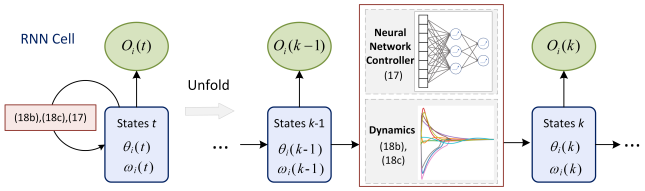


Fig. 4. Structure of RNN for the frequency control problem

The operation of RNN is shown by the left side of Fig. 4. 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. 4. 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 (17b) (17c) 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 (16) to satisfy inequality constraints. The output $O_i(k) = \begin{bmatrix} O_i^1(k) & O_i^2(k) \end{bmatrix}$ 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 (17a) as:

$$Loss = \sum_{i=1}^{N} \max_{k=0,\cdots,K} |O_i^1(k)| + \gamma \frac{1}{K}\sum_{k=1}^{K} O_i^2(k) \tag{18}$$

The trainable variables $\boldsymbol{\varphi}$ is specified in the neural network controller (16) and updated by gradient descent through the Loss function (18). 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 [34].

C. 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 (14)-(16) . 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 (14) and (15), 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 (14) and (15), 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. 4. Adam algorithm is adopted to update weights in each episode.

V. Simulation Studies

Case studies are conducted on the IEEE New England 10-machine 39-bus (NE39) power networks to illustrate the effectiveness of the proposed method. To ensure that our results apply in practice, simulations are conducted on the system with 6^{th}-order generator model as well as dynamic models for inverter-connected resources [20], [35], [36]. Firstly, we show that the proposed Lyapunov-based approaches for designing neural network controller can guarantee stability,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2022.3176525, IEEE Transactions on Power Systems

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 (17)

Input: The bound of $\overline{\theta}_i$ and $\overline{\omega}_i$ to generate the initial states *Initialisation* :Initial weights $\boldsymbol{\varphi}$ for control network

1: **for** $episode = 1$ to I **do**
2: Generate initial states $\theta_i^h(0), \omega_i^h(0)$ for the i-th bus in the h-th batch, $i = 1, \cdots, n$, $h = 1, \cdots, H$
3: Reset the state of cells in each batch as the initial value $x_i^h \leftarrow \{\theta_i^h(0), \omega_i^h(0)\}$.
4: RNN cells compute through K stages to obtain the output $\{O_{h,i}(0), O_{h,i}(1), \cdots, O_{h,i}(K)\}$
5: Calculate total loss of all the batches $Loss = \frac{1}{H}\sum_{h=1}^{H}\sum_{i=1}^{N} \max_{k=0,\cdots,K} |O_{h,i}^1(k)| + \gamma \frac{1}{K}\sum_{k=1}^{K} O_{h,i}^2(k)$
6: Update weights in the neural network by passing $Loss$ to Adam optimizer: $\boldsymbol{\varphi} \leftarrow \boldsymbol{\varphi} - \alpha \text{Adam}(Loss)$
7: **end for**

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.

A. Practical Implementation of Inverter-based Controllers

In this subsection, we show how the controllers $u_{\varphi_i}(\omega_i)$'s can be utilized in different types of inverter-connected resources in practice using the Western Electricity Coordinating Council (WECC) generic models as an example [36]. This system has been widely used for studying system response to electrical disturbances, including major disturbances such as loss of generation or large step change of load [37], [38]. The generic model is shown in Figure. 5 and consists of three modules:

1) The renewable energy generator/inverter model (Regc), which has inputs of real (Ipcmd) and reactive (Iqcmd) current command and outputs of real (Ip) and reactive (Iq) current injection into the grid model.
2) The renewable energy electrical controls model (Reec), which has inputs of real/reactive power setpoints (p_{ext}/q_{ext}) that can be externally controlled. The outputs are the real and reactive current command calculated according to the p_{ext}/q_{ext} and the voltage v_t.
3) Renewable energy plant controller (Repc) that determines active and reactive setpoints. The f/p (frequency/active power output) block emulates active power control, and v/q (voltage and reactive power output) block emulates volt/var control at the plant level.

Our paper essentially updates the f/p control block (red part in Fig. 5). Instead of using a linear droop controller, we design nonlinear control law $u(\omega_{\text{pll}})$ to replace existing linear f/p control law. The control of wind turbine generator (WTG) and solar PV can be compared using this generic model. For example, Fig. 6 shows the dynamic representation of large-scale PV plants built based on modules in Fig. 5 [36].

The control law $u(\omega_{\text{pll}})$ obtained from the proposed method serves as external command for the adjustment of active power setpoint in the module of inverters for Solar PV. Similarly, the control law $u(\omega_{\text{pll}})$ can also be used for WTG by replacing the block of f/p droop control. Detailed dynamic representation of WTG using the WECC generic blocks can be find in [36].

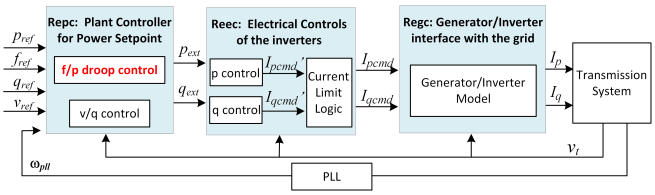


Fig. 5. Block diagram of WECC generic model [36]

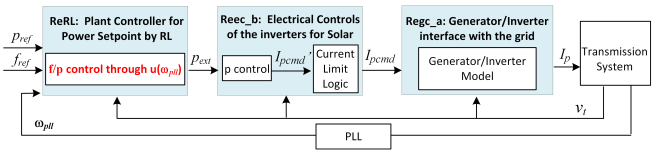


Fig. 6. WECC generic model for PV [36]

B. 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. ANDES (an open source package for power system dynamic simulation) is utilized to simulate the dynamic response from WECC generic model for solar PV and Type-4 wind turbine generation (WTG) as the renewable resources, and 6^{th}-order generator model with turbine-governing systems [36], [39]. We set 30% of active power generation from PV or WTG and the remaining 70% comes from synchronous generator. Parameters for the PV and WTG follow the default values in ANDES [39]. For training the neural network controller, the system is in the Kron reduced form [7], [40] and its dynamics is represented by (1). The bound on action $\overline{u}_i$ is generated to be uniformly distributed in $[0.8p_i, p_i]$. We generate the trajectories by randomly picking at most three generators to have a step load change uniformly distributed in $[-1, 1]$ p.u. We use a non-quadratic cost $C_i(\boldsymbol{u}_i) = \sum_{t=1}^{T} |u_i(t)|$ from [28] for case studies. The cost coefficient $\gamma = 0.002$. The stepsize between time states is set as $\Delta t = 0.01s$ and the total time stages is $K = 200$.

Since the power output of both PV and WTG follow the power setpoint accurately (numerical validation is provided in Appendix E), the system dynamics with PV or WTG as the renewables will be very similar. In the following part of this section, we show the simulation results using PV as the inverter-connected resources. 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:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2022.3176525, IEEE Transactions on Power Systems

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, 600 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 solving (17) using 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 [14]. It differs with 1) only in the training methods. 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. To encourage exploration, a zero-mean Gaussian noise is added to the control policy.

C. Necessity of Lyapunov-based Approach

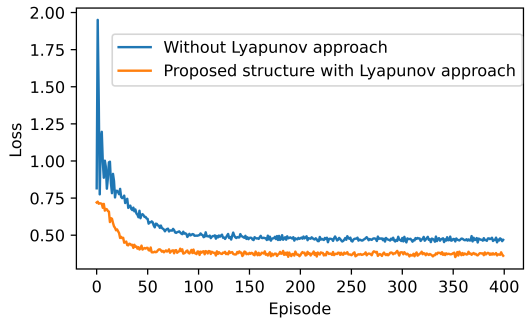
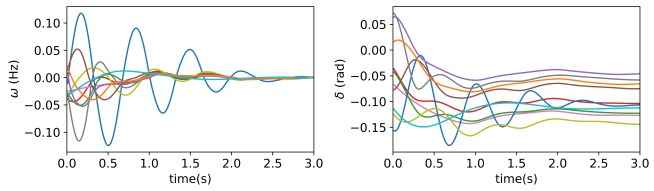


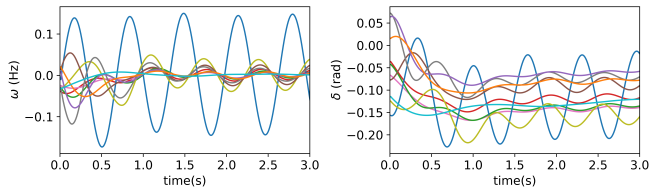
Fig. 7. Average batch loss along episodes for neural network controller designed with and without the Lyapunov-based approach. Both converges, with the former converging much for quickly than the latter.

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.

Fig. 7 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. 8b). In contrast, the controller constrained by the Lyapunov condition shows good performance (Fig. 8a). 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.



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



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

Fig. 8. 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. 7. The controller designed without the Lyapunov-based approach leads to unstable trajectories of the system.

D. 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. Fig. 9 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/generation. Suppose the load at bus 29 experiences a step load increase of 3 p.u. occurring at $t = 0.5$s. Figure 10 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 deviations. Therefore, the proposed RNN-Lyapunov approach has the minimal cost. The computational time of the proposed RNN based method is 1465.58s, while the computational time of REINFORCE policy gradient takes 5050.12s. Therefore, the proposed RNN based structure reduces computational time by approximate 70.98% compared with the general RL structure. The key reason for the better performance of RNN lies in the efficient usage of the physical model. For the model-free RL

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2022.3176525, IEEE Transactions on Power Systems

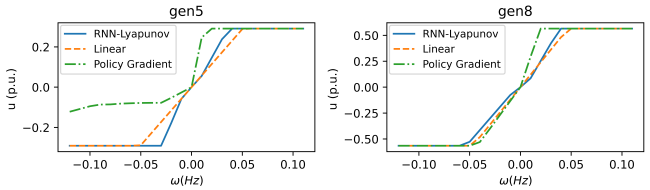
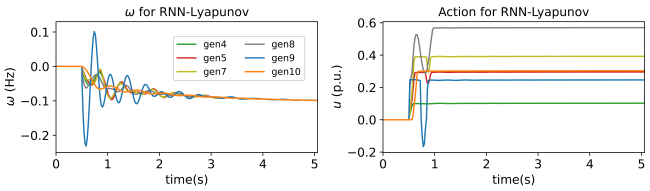
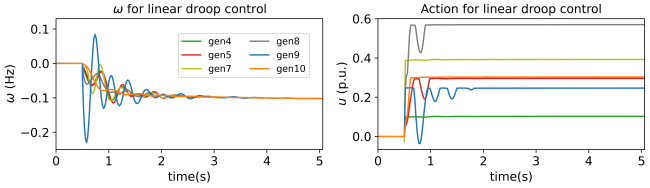


Fig. 9. Examples of learned controller u corresponding to RNN-Lyapunov, Linear droop control and Policy Gradient for generator buses 5 and 8. The comparison shows that the proposed Stacked-ReLU neural network learns nonlinear controllers in flexible shapes.

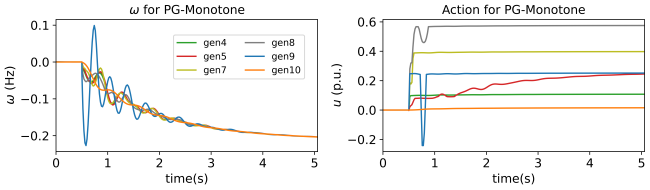
methods including REINFORCE policy gradient, it is well-known that the neural network will easily get stuck in the current weights based on the history trajectories [14]. This makes PG controller learn much smaller range of u (as shown in Fig. 9) since the training of neural network get stuck in the current weights without updating further. Moreover, to encourage the training of RL to explore some possibly better actions, the control policies in REINFORCE policy gradient are parameterized as Gaussian policies that add Gaussian noises to the implemented actions. Then, the gradient descent with respect to the weights in the neural network is calculated from a Gaussian distribution instead of a deterministic control law. This increases the computational burden and therefore causes the significant increase in their computational time.



(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

Fig. 10. 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. (c) Lyapunov-guided neural network controller learned from Policy Gradient with Monotone structure design. The proposed RNN controller has the smallest cost.

Next, we randomize the step load changes to simulate and test the performance of the three methods under multiple different trajectories. We randomly select three generators to let the step load change uniformly distributed in $U[-\Delta\bar{p}_l, \Delta\bar{p}_l]$, where $\Delta\bar{p}_l$ denotes the variation bound of the step load change. The average loss corresponding to $\Delta\bar{p}_l = 0.2, 0.4, \cdots, 1.4$ p.u. are illustrated in Fig. 11. Overall, the average loss in linear droop control and PG-Monotone is approximately 12.63% and 7.83% higher than RNN-Lyapunov trained with 600 trajectories, respectively. If reducing the number of training trajectories to 300, the loss will be 3.36% higher than that trained with 600 trajectories but still better than linear droop control and PG-Monotone. Therefore, the proposed method learn the nonlinear controller that leads to better average control performance under different scenarios. Moreover, it is generalizable to new scenarios even when there are not many trajectories for training.

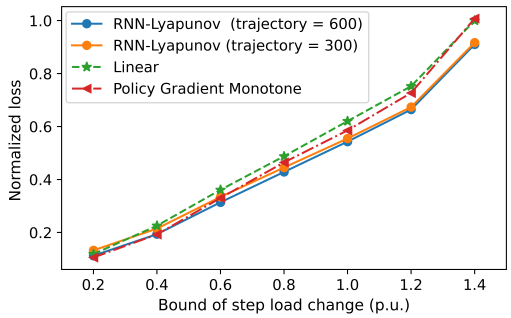


Fig. 11. Loss with different variation range of step load changes for RNN-Lyapunov (with the number of training trajectories to be 600 and 300, respectively), Linear droop controller and Policy Gradient. Compared with Linear droop controller and PG-Monotone, RNN-Lyapunov (trajectory=600) reduces the loss by approximately 12.63% and 7.83%, respectively.

VI. Conclusion

This paper 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.

References

[1] 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.

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

[3] B. K. Poolla, S. Bolognani, and F. Dorfler, "Optimal placement of virtual inertia in power grids," *IEEE Transactions on Automatic Control*, vol. 62, no. 12, pp. 6209–6220, 2017.

[4] 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*, vol. 35, no. 6, pp. 4383–4395, 2020.

[5] 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.

[6] 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.

[7] A. Ademola-Idowu and B. Zhang, "Frequency stability using inverter power control in low-inertia power systems," *IEEE Transactions on Power Systems*, vol. 36, no. 2, pp. 1628–1637, 2020.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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*, vol. 11, no. 2, pp. 1171–1182, 2019.

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

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

[16] 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.

[17] L. Guo, C. Zhao, and S. H. Low, "Graph laplacian spectrum and primary frequency regulation," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 158–165.

[18] C. Zhao, U. Topcu, N. Li, and S. Low, "Power system dynamics as primal-dual algorithm for optimal load control," *arXiv preprint arXiv:1305.0585*, 2013.

[19] A. Delavari and I. Kamwa, "Sparse and resilient hierarchical direct load control for primary frequency response improvement and inter-area oscillations damping," *IEEE Transactions on Power Systems*, vol. 33, no. 5, pp. 5309–5318, 2018.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] Y. Jiang, E. Cohn, P. Vorobev, and E. Mallada, "Storage-based frequency shaping control," *IEEE Transactions on Power Systems*, vol. 36, no. 6, pp. 5006–5019, 2021.

[25] V. Purba, B. B. Johnson, M. Rodriguez, S. Jafarpour, F. Bullo, and S. V. Dhople, "Reduced-order aggregate model for parallel-connected single-phase inverters," *IEEE Transactions on Energy Conversion*, vol. 34, no. 2, pp. 824–837, 2018.

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

[27] Y. Jiang, R. Pates, and E. Mallada, "Dynamic droop control in low-inertia power systems," *IEEE Transactions on Automatic Control*, vol. 66, no. 8, pp. 3518–3533, 2020.

[28] 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.

[29] B. Xu, J. Zhao, T. Zheng, E. Litvinov, and D. S. Kirschen, "Factoring the cycle aging cost of batteries participating in electricity markets," *IEEE Transactions on Power Systems*, vol. 33, no. 2, pp. 2248–2259, 2017.

[30] 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, 2018.

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

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

[33] 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.

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

[35] 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.

[36] E. Farantatos, "Model user guide for generic renewable energy system models," *Technical Update 3002014083, EPRI*, 2018.

[37] J. F. Hauer, W. A. Mittelstadt, K. E. Martin, J. W. Burns, H. Lee, J. W. Pierre, and D. J. Trudnowski, "Use of the wecc wams in wide-area probing tests for validation of system performance and modeling," *IEEE Transactions on Power Systems*, vol. 24, no. 1, pp. 250–257, 2009.

[38] G. Hou and V. Vittal, "Cluster computing-based trajectory sensitivity analysis application to the wecc system," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 502–509, 2011.

[39] H. Cui, F. Li, and K. Tomsovic, "Hybrid symbolic-numeric framework for power system modeling and analysis," *IEEE Transactions on Power Systems*, vol. 36, no. 2, pp. 1373–1384, 2020.

[40] 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.

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

APPENDIX A
PROOF OF LEMMA 2

Proof. The proof is similar to the one of [30, Lemma 14], which bounds $V(\boldsymbol{\delta}, \boldsymbol{\omega})$ term by term. Firstly, using the Rayleigh-Ritz theorem [41], 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 [31, Lemma 4], the potential energy term $W_{\mathrm{p}}(\boldsymbol{\delta})$ in (9) 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 [31, Lemma 4] and the Rayleigh-Ritz theorem, with some $\gamma_2 > 0$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2022.3176525, IEEE Transactions on Power Systems

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 (8) with

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

for sufficiently small $\epsilon>0$. □

APPENDIX B
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.,

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

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

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

which is exactly (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 (5a).

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. □

APPENDIX C
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

$$|y(x)-r(x)|\leq|r((k+1)\beta)-r(k\beta)| \tag{20}$$

By mean value theorem, we know that

$$r((k+1)\beta)-r(k\beta)=\beta\frac{\partial r(c)}{\partial x} \tag{21}$$

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 (14)(15). 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 (14) 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 D
FREQUENCY MEASUREMENT

We conduct simulations to confirm that the frequency measured by the PLL on the power electronic interfaces are aligned with the rotor angles on the synchronous generators.

Fig. 12 compares the rotor angle speed and the bus frequency measurement from PLL on different selected buses. Even though the shape of the frequency measurement alters a bit when there is a large spike in the rotor angle speed (shown as the blue curve), the sign of the measurement is always the same as the rotor angle speed. Hence, the designed controller is still the same sign with the rotor angle speed, which guarantees that the system is asymptotically stable. Moreover, the frequency measurement generally follows the rotor angle speed closely and achieves the same steady-state values.

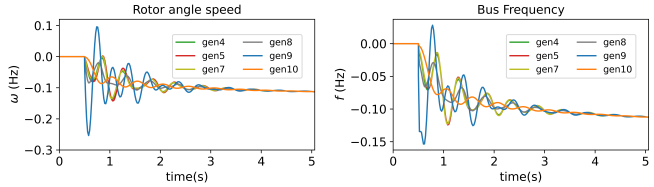


Fig. 12. Comparison of rotor angle speed and the bus frequency measurement from PLL.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2022.3176525, IEEE Transactions on Power Systems

APPENDIX E
COMPARISON OF PV AND WTG

To validates that our approach is practical and applicable to different types of resources, Fig. 13 compares the rotor dynamics ω at selected buses when wind or solar are employed as the renewable resource. After the step change in load, the frequency behavior and the control action under the learned RNN-Lyapunov controllers are similar for wind and solar (with the frequency deviations with solar PV being slightly smaller).

Fig. 14 compares the active power output from WTG/PV and the setpoints from the control law u. All the values have subtracted their steady state value p_{ref}, and therefore the active power from renewables can be negative. Both power outputs follow the power setpoints very closely. This justifies that the dynamics of the inverter-connected resources are much faster than the simulation time step for frequency regulation. Although the specific model for WTG and PV are different, we can design the same control low to adjust the active power setpoints for primary frequency regulation.

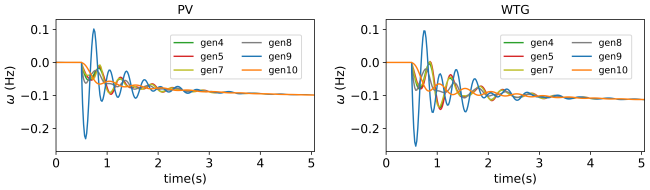
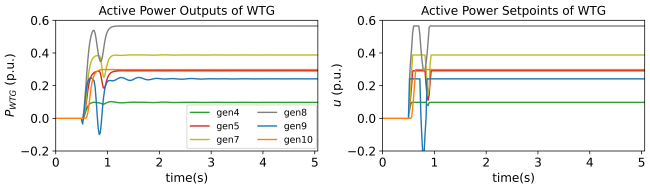
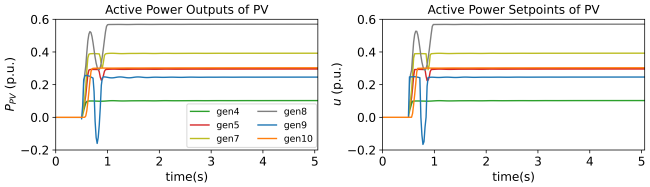


Fig. 13. Dynamics of the frequency deviation w of (left) PV and (right) WTG in selected buses. The frequency deviations are very similar when two different resources are used.



(a) Active power output and setpoints from WTG

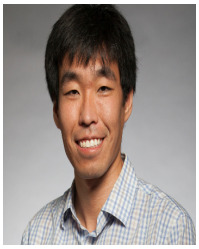


(b) Active power output and setpoints from solar PV

Fig. 14. The active power output and setpoints in selected generator buses corresponding to (a) WTG. (b) PV. The power output follows the power setpoint accurately with almost no delays.

Wenqi Cui received the B.Eng. degree and M.S. degree in electrical engineering from Southeast University, Nanjing, China, and Zhejiang University, Hangzhou, China, in 2016 and 2019, respectively. She is currently working toward the Ph.D. degree in Electrical Engineering at the University of Washington, Seattle, WA, USA. She works on control, optimization, and machine learning, with applications in cyberphysical systems.

Yan Jiang received the B.Eng. degree in electrical engineering and automation from Harbin Institute of Technology, Harbin, CHN, in 2013, the M.S. degree in electrical engineering from Huazhong University of Science and Technology, Wuhan, CHN, in 2016, and the Ph.D. degree in electrical engineering with the M.S.E. degree in Applied Mathematics and Statistics from Johns Hopkins University, Baltimore, USA, in 2021. She is currently a Postdoctoral Scholar with the Department of Electrical and Computer Engineering at University of Washington, Seattle, USA. Her research interests lie in the area of control of power systems.



Baosen Zhang received his Bachelor of Applied Science in Engineering Science degree from the University of Toronto in 2008; and his PhD degree in Electrical Engineering and Computer Sciences from University of California, Berkeley in 2013. He was a Postdoctoral Scholar at Stanford University. He is currently an Associated Professor in Electrical and Computer Engineering at the University of Washington, Seattle, WA. His research interests are in control, optimization and learning applied to power systems and other cyberphysical systems. He received the NSF CAREER award as well as several best paper awards.