Adaptive Neurocontrol for Grid-Following Inverters

Chin-Yao Chang and Yashen Lin

National Renewable Energy Laboratory

NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Technical Report
NREL/TP-5D00-83921
October 2022

Adaptive Neurocontrol for Grid-Following Inverters

Chin-Yao Chang and Yashen Lin

National Renewable Energy Laboratory

Suggested Citation
Chang, Chin-Yao and Yashen Lin. 2022. *Adaptive Neurocontrol for Grid-Following Inverters*. Golden, CO: National Renewable Energy Laboratory. NREL/TP-5D00-83921. https://www.nrel.gov/docs/fy23osti/83921.pdf.

NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Technical Report
NREL/TP-5D00-83921
October 2022

National Renewable Energy Laboratory
15013 Denver West Parkway
Golden, CO 80401
303-275-3000 • www.nrel.gov

NOTICE

This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the U.S. Department of Energy Office of Electricity Delivery and Energy Reliability. The views expressed herein do not necessarily represent the views of the DOE or the U.S. Government.

Executive Summary

The new generation of power systems involve many autonomous operating units with the flow of data being restricted by privacy concerns or infrastructure hurdles. The systems are also time-varying, so the control should adjust in a timely manner to avoid failures that usually have very negative economical implications. Adaptive neurocontrol takes elements from adaptive control (great for time-varying problems) and model identification (could be in the form of neural network) by using local available data. Those properties make adaptive neurocontrol a compelling tool for power system applications. This work is about leveraging adaptive neurocontrol for decentralized control of grid-following inverters. Some encouraging results are found and we expect that adaptive neurocontrol can also work on many other online power system control problems.

1 Adaptive Neurocontroller Design

Adaptive neurocontrol is considered particularly suitable for controlling highly uncertain, nonlinear, and complex systems Ge and C. Wang 2004; Ge and J. Wang 2002. However, the complexity involved in the associated Lyapunov stability analyses restricts the majority of the existing adaptive neurocontroller to single-input single-output (SISO) systems, which have relatively narrow applications. In this brief report, we make minor extensions of the contents in Lavretsky 2008 for adaptive neurocontrol for multiple-input multiple-output (MIMO) systems.

We consider the following control problem:

$$\begin{aligned} \dot{x} &= Ax + B\Lambda(u - f(x,u)), \\ y &= Cx, \end{aligned} \tag{1.1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ and $y \in \mathbb{R}^s$. We assume that A, B, C are known for the controller, and the diagonal matrix Λ and $f(x,u)$ are unknown. The objective is to track the output y of (1.1) with y_m generated by the following reference dynamical system:

$$\begin{aligned} \dot{x}_m &= A_m x_m + B_m r, \\ y_m &= C x_m. \end{aligned} \tag{1.2}$$

The adaptive feedback controller of (1.1) is formulated as:

$$u = K_y^\top y + K_r^\top r + \Theta^\top \Phi(y,u), \tag{1.3a}$$

$$\dot{K}_y = g_y(K_y, y, y_m), \tag{1.3b}$$

$$\dot{K}_r = g_r(K_r, y, y_m, r), \tag{1.3c}$$

$$\dot{\Theta} = g_\Phi(\Theta, y, y_m, \Phi(y,u)), \tag{1.3d}$$

where $\Theta^\top \Phi(y,u)$ is an approximation (identified by data) of $f(x,u)$, g_y, g_r, and g_Φ are designed based on Lyapunov analysis. The formulation above is quite general in the sense that in many applications, we can essentially put everything nonlinear/unknown to $f(x,u)$ and use whatever available data to best approximate $f(x,u)$. Intuitively, the last term of (1.3a), $\Theta^\top \Phi(y,u)$, is mostly about canceling out the nonlinear/unknown term $f(x,u)$. As will become clear later, the controller performance is directly dependent on the accuracy of the approximation of $f(x,u)$.

In the following, we show step-by-step how g_y, g_r, and g_Φ are designed and the resulting stability results. The key is identifying the dynamics of the tracking error $e(t) := y(t) - y_m(t)$ and designing a controller that can stabilize such dynamics. Assumption 1 paves the possibility of the existence of a controller such that steers $e(t)$ to the origin.

Assumption 1. ***(Existence of exact matching dynamics).*** *There exist $\hat{K}_y$, $\hat{K}_u$, $\hat{\Theta}$, ε_f, and a finite scalar ε such that*

$$C(A + B\Lambda \hat{K}_y^\top C) = CA_m, \tag{1.4a}$$

$$CB\Lambda \hat{K}_r^\top = CB_m, \tag{1.4b}$$

$$\hat{\Theta}^\top \Phi(y,u) - f(x,u) = \varepsilon_f, \tag{1.4c}$$

$$\|\varepsilon_f\| < \varepsilon. \tag{1.4d}$$

Equations (1.4a) and (1.4b) ensure that (1.2) can be matched by the linear part of a controlled (1.1). Equations (1.4c) and (1.4d) bound the error of the estimation on the nonlinear function $f(x,u)$. If Assumption 1 holds and $\varepsilon = 0$, then the adaptive neurocontroller explained later can match the actual system dynamics (1.1) with the targeted one (1.2) perfectly. Assumption 1 also helps simplify the error dynamics as shown in the following:

$$\begin{aligned} \dot{e} &= \dot{y} - \dot{y}_m = C\dot{x} - C\dot{x}_m \\ &= C(A + B\Lambda K_y^\top C)x + CB\Lambda K_r^\top r + CB\Lambda(\Theta^\top \Phi(y,u) - f(x,u)) - CA_m x_m - CB_m r \\ &= CA_m(x - x_m) + CB\Lambda(K_y^\top - \hat{K}_y^\top)y + CB\Lambda(K_r^\top - \hat{K}_r^\top)r + CB\Lambda(\Theta^\top - \hat{\Theta}^\top)\Phi(y,u) - CB\Lambda\varepsilon_f. \end{aligned}$$

Defining $e_x = x - x_m$, $\Delta K_y = K_y - \hat{K}_y$, $\Delta K_r = K_r - \hat{K}_r$ and $\Delta\Theta = \Theta - \hat{\Theta}$, we rewrite the equation above as:

$$\dot{e} = CA_m e_x + CB\Lambda(\Delta K_y^\top y + \Delta K_r^\top r + \Delta\Theta^\top \Phi(y,u) - \varepsilon_f). \tag{1.5}$$

We next define the Lyapunov function in the following:

$$V(e,\Delta K_y,\Delta K_r,\Delta\Theta) = e^\top Pe + \mathrm{tr}(\Delta K_y^\top \Gamma_y^{-1}\Delta K_y|\Lambda|) + \mathrm{tr}(\Delta K_r^\top \Gamma_r^{-1}\Delta K_r|\Lambda|) + \mathrm{tr}(\Delta\Theta^\top\Gamma_\Theta^{-1}\Delta\Theta|\Lambda|), \tag{1.6}$$

where P, Γ_y, Γ_e and Γ_Θ are positive definite matrices with proper dimensions. Using (1.5), we write the time derivative of the Lyapunov function as:

$$\begin{aligned}\dot V = {} & e_x^\top(A_m^\top C^\top PC + C^\top PCA_m)e_x + 2e^\top PCB\Lambda(\Delta K_y^\top y + \Delta K_r^\top r + \Delta\Theta^\top\Phi(y,u) - \varepsilon_f) \\ & + 2\,\mathrm{tr}(\Delta K_y^\top\Gamma_y^{-1}\dot K_y|\Lambda|) + 2\,\mathrm{tr}(\Delta K_r^\top\Gamma_r^{-1}\dot K_r|\Lambda|) + 2\,\mathrm{tr}(\Delta\Theta^\top\Gamma_\Theta^{-1}\dot\Theta|\Lambda|).\end{aligned} \tag{1.7}$$

Using the property that $a^\top b = \mathrm{tr}(ba^\top)$, we rewrite (1.7) as:

$$\begin{aligned}\dot V = {} & e_x^\top(A_m^\top C^\top PC + C^\top PCA_m)e_x - 2e^\top PCB\Lambda\varepsilon_f \\ & + 2\,\mathrm{tr}(\Delta K_y^\top(\Gamma_y^{-1}\dot K_y + ye^\top PCB\cdot\mathrm{sgn}(\Lambda))|\Lambda|) \\ & + 2\,\mathrm{tr}(\Delta K_r^\top(\Gamma_r^{-1}\dot K_r + re^\top PCB\cdot\mathrm{sgn}(\Lambda))|\Lambda|) \\ & + 2\,\mathrm{tr}(\Delta\Theta^\top(\Gamma_\Theta^{-1}\dot\Theta + \Phi(y,u)e^\top PCB\cdot\mathrm{sgn}(\Lambda))|\Lambda|).\end{aligned} \tag{1.8}$$

Now we assume that there exists $Q \succeq 0$ such that $A_m^\top C^\top PC + C^\top PCA_m = -Q$. In addition, we define g_y, g_r, g_Φ respectively as:

$$\dot K_y = g_y(K_y,y) = -\Gamma_y ye^\top PCB\cdot\mathrm{sgn}(\Lambda) \tag{1.9a}$$

$$\dot K_r = g_r(K_r,y,r) = -\Gamma_r re^\top PCB\cdot\mathrm{sgn}(\Lambda), \tag{1.9b}$$

$$\dot\Theta = g_\Phi(\Theta,y,\Phi(y,u)) = -\Gamma_\Theta\Phi(y,u)e^\top PCB\cdot\mathrm{sgn}(\Lambda). \tag{1.9c}$$

The controller gain dynamics defined above make the last three terms of $\dot V$ of (1.8) zero. Therefore, substituting $A_m^\top C^\top PC + C^\top PCA_m = -Q$ and (1.9) back to (1.8), we conclude that $\dot V < 0$ if

$$\begin{aligned} & -e_x^\top Qe_x - 2(e_x^\top C^\top)PCB\Lambda\varepsilon_f < 0 \\ \Longrightarrow & \|2(e_x^\top C^\top)PCB\Lambda\varepsilon_f\| < \|e_x^\top Qe_x\| \\ \Longrightarrow & \frac{2\|C^\top PCB\Lambda\|\varepsilon}{\lambda_{\min}(Q)} < \|e_x\|. \end{aligned} \tag{1.10}$$

By Lyapunov theory, we conclude that under the adaptive neurocontrol (1.9), the error converges to a compact set $\left\{e_x : \|e_x\| \le \frac{2\|C^\top PCB\Lambda\|\varepsilon}{\lambda_{\min}(Q)}\right\}$. Furthermore, if the estimation of the nonlinear function $f(x,u)$ is exact by $\hat\Theta^\top\Phi(y,u)$, i.e., $\varepsilon = 0$, we conclude that the adaptive neurocontrol (1.9) can bring the error to zero because $\|e_x\| \le \frac{2\|C^\top PCB\Lambda\|\varepsilon}{\lambda_{\min}(Q)} = 0$.

2 Numerical Results

The numerical studies are divided into two parts. First, we randomly generate some numerical examples for the output feedback control with online learning of the unknown/nonlinear part of the system dynamics to test the adaptive neurocontroller. The second part takes numerical setup from the IEEE 39 bus system with some of the generators being replaced by grid-following inverters. The adaptive neurocontrol is applied for decentralized control of the inverters.

2.1 Randomly Generated Systems with Online Learning

We consider a dynamical system in the form of (1.1) with $x \in \mathbb{R}^5$, $u \in \mathbb{R}^4$, $y \in \mathbb{R}^4$. The linear parts of the actual and targeted dynamics are set as the same, namely, $A = A_m$, $B\Lambda = B_m$. Every element of B_m and C are randomly chosen between 0 and 1; A_m is a diagonal matrix with the diagonal elements being all 0.98 for the stability. The discrepancy between actual and the targeted ones comes from $f(y)$, given as:

$$f(y) = f_c \cdot \cos(W_c \cdot y) + f_s \cdot \sin(W_s \cdot y), \tag{2.1}$$

where all the elements of f_c, W_c, f_s and W_s are randomly generated with the values in the interval $[0,1]$. Note that (2.1) is structured such that there exists $\hat{\Theta}$ and a finite dimensional $\Phi(y)$ such that $\hat{\Theta}\Phi(y) = f(y)$. The linear system generated this way has Assumption 1 hold, so the adaptive neurocontrol is expected to work well in this setup.

In the context of adaptive neurocontroller, the approximation of $f(y)$ lies in two main components, Θ and $\Phi(y)$, where Θ is updated through the adaptive controller g_Φ and the parameterization of $\Phi(y)$ is given offline and supposedly trained by offline data. Here we assume the controller does not know that $f(y)$ is structured as sinusoidal functions of y, so generic neural network for Φ is applied, where the open source machine learning platform, Tensorflow *TensorFlow: An end-to-end open source machine learning platform,* is applied. We set the number of hidden layers as 3 and number of neurons for every layer as 60 for the Tensorflow model. Instead of generating offline data to initialize $\Phi(y)$, we train $\Phi(y)$ online with realtime data.

All the variables are initialized at zeros except x_m and x, where each element of x_m takes random value in the interval $[0,1]$ and $x = x_m + w_m$ with w_m being the noises which take values in $[0, 0.2]$. The simulation of in this section involves Euler discretization of (1.1), (1.2), and (1.9) with step size 10^{-5} for simplicity, which can be viewed as good approximation of the continuous time version given that the step size is very small. Because in the first few iterations, $\Theta \cdot \Phi(y)$ could be far off from $f(y)$, we adjust the controller u in (1.3) as shown in the following for the first 100 iterations:

$$\begin{aligned} u &= K_y^\top y + K_r^\top r, \\ \dot{K}_y &= g_y(K_y, y), \\ \dot{K}_r &= g_r(K_r, y, r), \end{aligned} \tag{2.2}$$

which essentially drops the Θ component. Starting from iteration 100, we compare the trajectories for controller (2.2) and (1.3). As shown in the left sub-figures of Figures 1 and 2, starting from iteration 100, the adaptive neurocontrol helps fill the gap between the difference between y and y_m (blue lines have lower $\|e\|$ compared to orange lines). However, at some point, the effect of overfitting kicks in so we see some transient spikes of between iterations 200-400. Eventually, $\|\Theta \cdot \Phi(y) - f(y)\|$ stabilize to some small values as shown in the left sub-figures of Figures 1 and 2, resulting to more ideal x for controller (1.3) compared to (2.2).

2.2 Control of Grid-Following Inverters

One of the challenges involved in replacing synchronous generators with inverter-interfaced energy resources is that the latter do not provide mechanical inertia as opposed to the former. Synchronous generators can naturally respond to power imbalance by rotating speed (translated to frequency deviations), while inverters just follow the preset values (power injections or voltage). Designing proper inverter controllers to maintain the stability of the overall system is necessary for future grids with high penetration of inverters.

We take the IEEE 39 bus system numerical setup in Lin et al. 2021 as shown in Figure 3. Each of the buses 30-39 has one generator connected. The generators for buses 31-35 are grid-following inverters, and the other generators

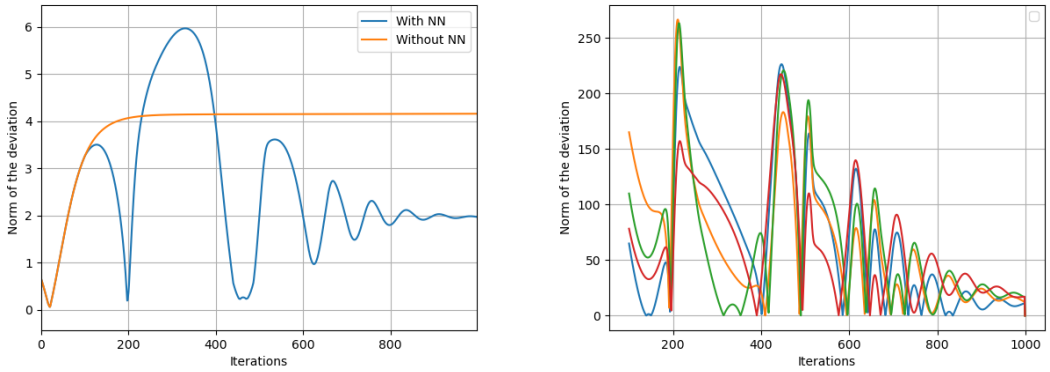

Figure 1. Test case 1. The figure on the left is $\|e\|$ and the figure on the right shows the norm of each element of $\Theta \cdot \Phi(y) - f(y)$.

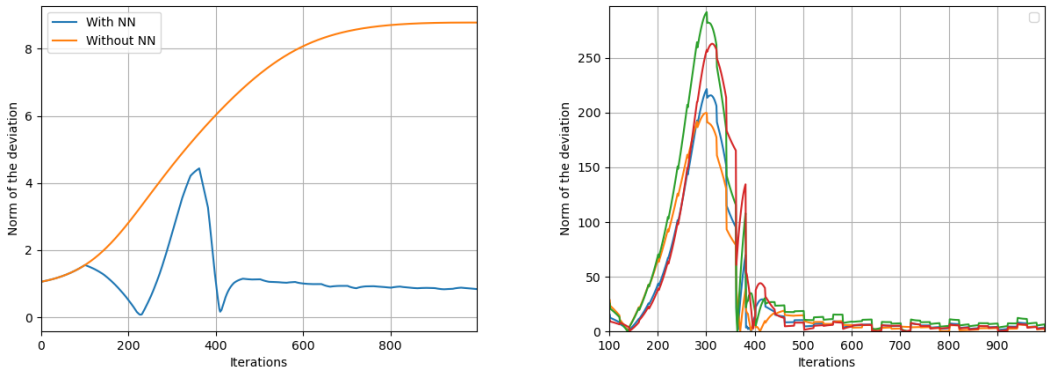

Figure 2. Test case 2. The figure on the left is $\|e\|$ and the figure on the right shows the norm of each element of $\Theta \cdot \Phi(y) - f(y)$.

are all synchronous generators. We follow standard PI controls for grid-following inverters as shown in Figure 4. The adaptive neurocontroller is next applied to adjust P_{ref} and Q_{ref}. For a given load profile, the overall system dynamics for the IEEE 39 bus system with grid-following inverters can be compactly written as:

$$\dot{x}_i = h_i(x_i) + B_i u_i - f_i(x, l), \quad \forall i \in \mathcal{G} = \{30, \cdots, 39\}, \tag{2.3}$$

where x_i collects all the variables involved in local control loops for generator i, $u_i = [P_{ref,i}, Q_{ref,i}]^\top$ if bus i is connected with a inverter-interfaced generator, $u_i = V_{ref,i}$ (voltage magnitude reference) for bus i if it is connected with a synchronous generator, h_i is the nonlinear function describing the local controller dynamics, B_i is a given matrix with proper dimensions, $f = [f_1, \cdots, f_{10}]^\top$ describes how the local generator variables are coupled, which essentially embedded the power flow equations. Note that f uses the Kron reduced power flow dynamics with a given load profile $l \in \mathbb{R}^{78}$, which collects the active and reactive loads for all the buses.

There are some hurdles to address before applying the adaptive neurocontroller to system (2.3). One is that the given control loop dynamics, $h_i(\cdot)$ for all i, are nonlinear. This can be addressed by rewriting the nonlinear model as a finite dimensional lifted linear model with proper selection of basis functions, written as:

$$\dot{\hat{x}}_i = A_i(\hat{x}_i) + B_i u_i - \hat{f}_i(\hat{x}, l), \quad \forall i \in \mathcal{G} = \{30, \cdots, 39\}. \tag{2.4}$$

The second challenge is the decentralized control architecture of the generators for the purpose of autonomy and flexibility. Each u_i in (2.4) can only use locally attainable measurements to generate, while the adaptive neurocontroller (1.9) require knowing all the state variables, $x = [x_1^\top, \cdots, x_{10}^\top]^\top$ to generate u (the y in (1.9) is replaced by x and $e = x - x_m$ instead of $(y - y_m)$ because C is essentially a identity matrix in (2.4)). This structural limitation of the information exchange makes a perfect decentralized controller for the entire grid fundamentally non-exist. We therefore take a step back and adjust the adaptive controller in an attempt to maximize the values of locally attainable

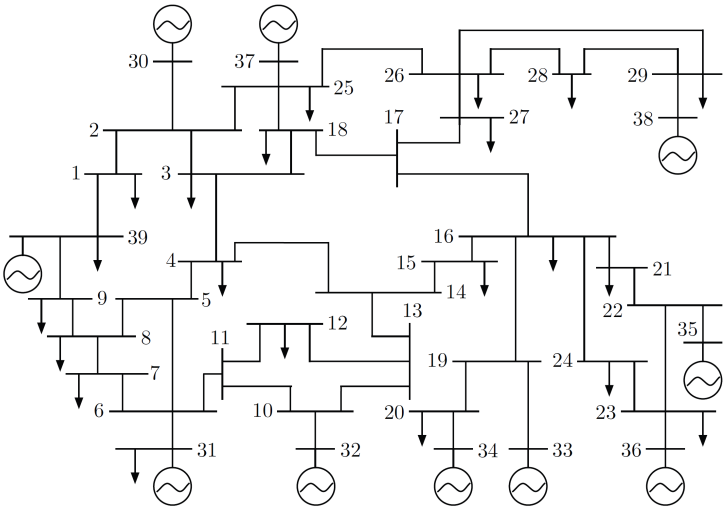

Figure 3. IEEE 39 bus system.

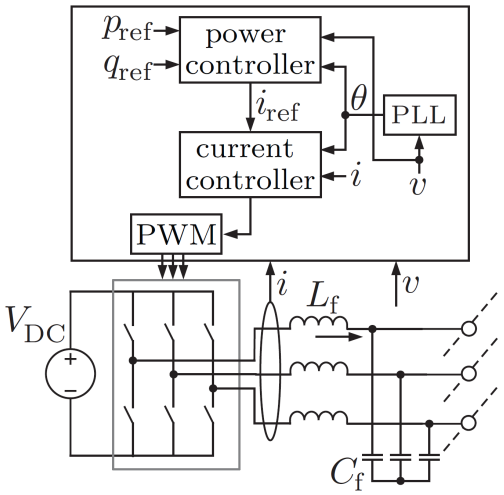

Figure 4. Control block diagram for grid-following inverters Lin et al. 2021.

information for each generator. Define $\hat{f}^\star(\cdot, l^\star)$ that is the concatenation of $\hat{f}_1^\star(\cdot, l^\star), \cdots, \hat{f}_{10}^\star(\cdot, l^\star)$ as the offline identified Kron reduced power flow dynamics and $l^\star$ is the estimated load profile for generating the Kron reduced power flow model. The target system dynamic is written as:

$$\dot{\hat{x}}_{m,i} = A_i \hat{x}_{m,i} + B_i u_{m,i} - \hat{f}_i^\star(\hat{x}_m, l^\star), \quad \forall i \in \mathscr{G} = \{30, \cdots, 39\}. \tag{2.5}$$

The actual system dynamics (2.4) and the targeted one (2.5) are identical except $\hat{f}$ and the load l. One of the main goals of the adaptive control u is then filling the gap between $\hat{f}_i(\cdot, l)$ and $\hat{f}_i^\star(\cdot, l^\star)$. As mentioned earlier in this section, local available information can not perfectly track the globally coupled component $\hat{f}$. However, globally measurable variable, frequency ω, provides insight of power imbalance of the grid and in turn can be helpful balancing out the gap between $\hat{f}_i(\cdot, l)$ and $\hat{f}_i^\star(\cdot, l^\star)$. Therefore, we adjust the adaptive controller (1.9) for this inverter controller application in the following:

$$\dot{K}_{x_i} = -\Gamma_{x_i} x_i e_i^\top B, \tag{2.6a}$$

$$\dot{\Theta}_i = -\Gamma_{\Theta_i} \mathbf{1} (\omega - \omega^\star) e_i^\top B, \tag{2.6b}$$

where Γ_{x_i} and Γ_{Θ_i} are chosen as $10^{-3} \cdot I$ for all $i = 1, \cdots, 10$. Note that $\Phi(\cdot, \cdot)$ in (1.9) is replaced by a simple linear function for the difference between measured frequency and the nominal one. A more sophisticated selection of Φ may exist, but we manage to have satisfying results with this simple selection of Φ.

In the first numerical example, we have the initialized $\hat{x} \neq x_m$ and $u \neq u_m$, while have $\hat{f}_i = \hat{f}_i^\star$ and $l = l^\star$. Unsurprisingly, the controller can bring back the frequency and the voltage magnitudes back to the targeted values, shown in Figure 5. The second example we add perturbations on the load profile such that the overall load is 1% higher than the estimated one ($l \neq l^\star$). As shown in Figure 6, we observe some deviation of both the frequency and voltage magnitudes, while it has less oscillatory trajectories compared to the case without control. The purpose of a decentralized controller is all about the stability, so some moderate deviations are acceptable. Secondary or tertiary controls with longer time frames can bring the operation points back to the ideal ones.

We conclude the report by noting that the main point of adaptive neurocontrol is regulating the state or output variables within a certain range of deviation. The range of deviation, however, is not pre-determined and highly dependent on how well the unknown nonlinear functions are approximated by either physical models or neural networks. The numerical results of grid-following controls reflect the point: though the controlled trajectories are smooth compared to non-controlled ones, the point that the system stabilizes at can deviate from the ideal values quite a bit. To achieve the goal of bringing down the deviation toward zero, some supplementary controls may be very-well needed on top of adaptive neurocontrol.

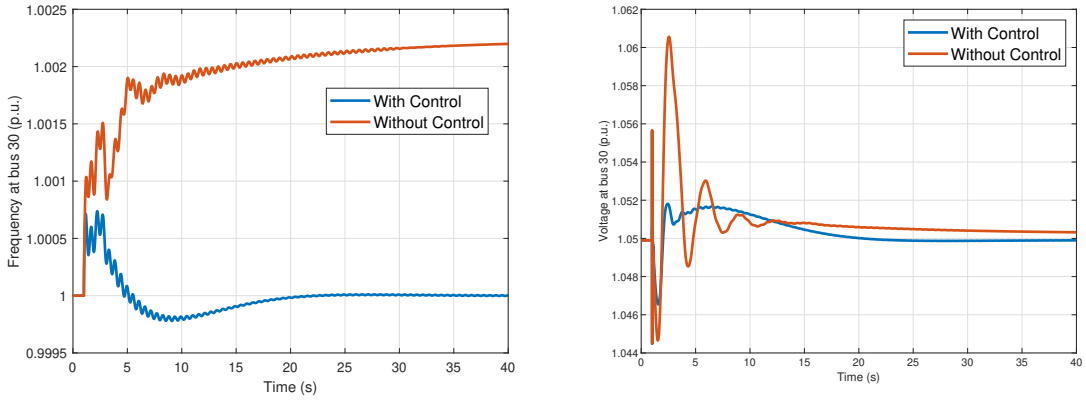

Figure 5. Simulation results for the scenario that the pre-identified $\hat{f}$ is accurate.

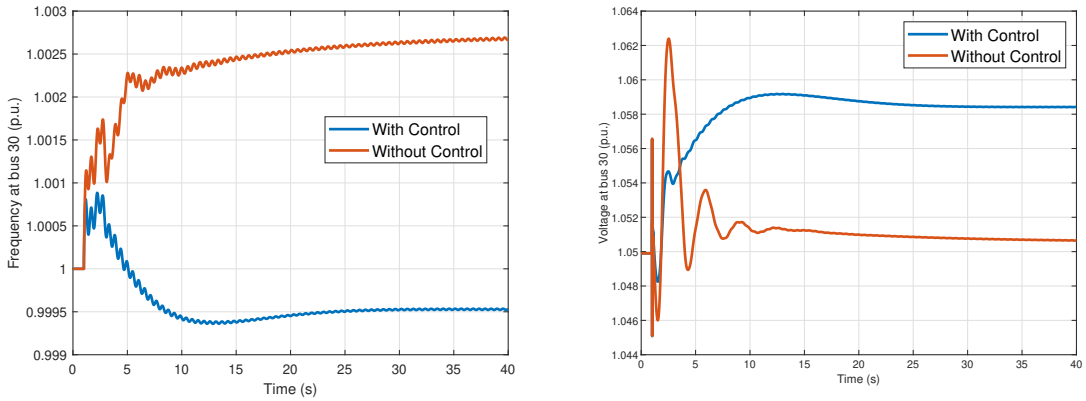

Figure 6. Simulation results for loads being perturbed 1%.

References

Ge, S. S., and C. Wang. 2004. "Adaptive neural control of uncertain MIMO nonlinear systems." *IEEE Transactions on Neural Networks* 15 (3): 674–692.

Ge, S. S., and J. Wang. 2002. "Robust adaptive neural control for a class of perturbed strict feedback nonlinear systems." *IEEE Transactions on Neural Networks* 13 (6): 1409–1419.

Lavretsky, E. 2008. "Adaptive control: Introduction, overview, and applications." In *Lecture notes from IEEE Robust and Adaptive Control Workshop.*

Lin, Y., G.-S. Seo, S. Vijayshankar, B. Johnson, and S. Dhople. 2021. "Impact of Increased Inverter-based Resources on Power System Small-signal Stability." In *2021 IEEE Power & Energy Society General Meeting (PESGM),* 01–05. IEEE.

TensorFlow: An end-to-end open source machine learning platform. https://www.tensorflow.org/. [Online; accessed 26-March-2022].