Improving primary frequency response in networked microgrid operations using multilayer perceptron-driven reinforcement learning

Nikitha Radhakrishnan\textsuperscript{1} \textsuperscript{,} Indrasis Chakraborty\textsuperscript{2}, Jing Xie\textsuperscript{2}, Priya Thekkumparambath Mana\textsuperscript{1}, Francis K. Tuffner\textsuperscript{2}, Bishnu P. Bhattarai\textsuperscript{1}, Kevin P. Schneider\textsuperscript{2}

\textsuperscript{1}Pacific Northwest National Laboratory, 902 Battelle Blvd., Richland, WA 99354, USA
\textsuperscript{2}Pacific Northwest National Laboratory, Seattle Research Center, Seattle, WA 98109, USA
E-mail: nikitha.r@pnnl.gov

Abstract: Individual microgrids can improve the reliability of power systems during extreme events, and networked microgrids can further improve efficiency through resource sharing and increase the resilience of critical end-use loads. However, networked microgrid operations can be subject to large transients due to switching and end-use loads, which can cause dynamic instability and lead to system collapse. These transients are especially prevalent in microgrids with high penetrations of grid-following inverter-connected renewable energy resources, which do not provide the system inertia or fast frequency response needed to mitigate the transients. One potential mitigation is to engage the existing generator controls to reduce system voltage in response to a frequency deviation, thereby reducing load and improving primary frequency response. This study investigates the use of a reinforcement-learning-based controller trained over several switching transient scenarios to modify generator controls during large frequency deviations. Compared to previously used proportional–integral controllers, the proposed controller can improve primary frequency response while adapting to changes in system topologies and conditions.

1 Introduction
A microgrid is a group of interconnected loads and distributed energy resources (DERs) within clearly defined electrical boundaries that act as a single controllable entity and can operate while connected to a grid or in autonomous/islanded mode. Microgrids can provide energy security to all critical facilities and services and can evolve as a key building block for the future power grid [1].

Individual microgrids offer resilience benefits when they can serve electricity demand during a bulk power system outage following extreme events [2–7]. The benefits of individual microgrids can be expanded by interconnecting multiple microgrids to share resources, especially if the bulk power system is unavailable for an extended period [8, 9]. Microgrids may be ‘networked’ through a connection with the distribution system after one or more normally-open switches are closed. The challenge with interconnecting microgrids is that switching operations between microgrids can result in large deviations in the frequency and voltage, and possible collapse of one or more of the microgrids [10–15].

One factor contributing to the large switching transients is insufficient system inertia in the power grid. Power system inertia, or inertial response, is a property of large synchronous generators, which contain large synchronous rotating masses; it acts to overcome the immediate imbalance between the power supply and demand in the electrical grid. Traditionally, in a system operating with only diesel generators, the rotating mass of the generators provide the required inertial response. However, to minimise emissions and take advantage of existing DERs like rooftop photovoltaic (PV), many of the available generation assets may be renewables (solar, wind etc.) that are interconnected with grid-following inverters, which provide no inertia or fast frequency response [16].

Additionally, during a switching transient, it may not be possible to maintain the system frequency within the normal operating ranges for inverters as prescribed in the IEEE Standard for Interconnecting Distributed Resources with Electric Power Systems, Amendment 1 (IEEE Standard 1547a™–2014) [17]. Operation outside these specified ranges can cause inverter-connected generation assets to trip off-line during the transient. As a solution, existing diesel generators are oversized to provide inertia and operated at reduced efficiencies, which increases capital and maintenance costs [18]. An alternative to the installation of additional or larger units is to increase the utilisation of the control systems on existing generators [19–21].

A method for improving primary frequency response by engaging the voltage regulation equipment on microgrid diesel generators was presented in [10]. That work established the concept of leveraging the conservation voltage reduction (CVR) effect to maximise the frequency nadir during a transient in a dynamic time frame. A proportional–integral (PI) controller was designed to adjust the desired voltage input to the automatic voltage regulator (AVR) of a diesel generator as a function of a drop in system frequency, reducing the voltage during the transient at the generator terminals. The reduced terminal voltage reduces the average system voltage, which can lead to a reduction in load and less electrical torque on rotating machines. The result is an improved primary frequency response, which improves resilience without the need to oversize existing generation units or to add more units.

The proof-of-concept work designed a PI controller to alter voltage reference values for an AVR as a function of frequency deviation from the desired value during a transient event. The PI controller was manually tuned and added to only one diesel generator of the test system. However, with multiple generators in microgrids, the coordinated tuning and engagement of multiple PI controllers pose additional challenges. Though PI controllers work well at the device level, at the system level they are highly sensitive to controller gain, which is difficult to assign when multiple PI controllers are incorporated [22, 23] in a connected network without extensive communication capabilities, such as in a microgrid. They also exhibit sluggish control response to sudden disturbances and are highly prone to oscillatory behaviours.

One way to address these drawbacks is to supplement the PI controller with a neural-network-based tuner that would adapt the...
controller gains according to the system conditions. Though effective in addressing traditional PI controller drawbacks, the additional controller would increase the controller’s response time and is more suited for supplementing existing control systems [24, 25]. Alternatively, previous works address these drawbacks in microturbines [26], hydrothermal systems [27], power quality devices [28], and motor speed and position control [29, 30] by deploying conventional PI controllers with neural-network-based controllers and observed better control system performance and stability.

This paper builds on the past work and explores using a neural-network-based control method – reinforcement learning (RL) – to modify set points of the existing voltage regulation equipment on diesel generators to mitigate the frequency deviations that occur in low-inertia microgrids during transients. This work proposes a RL-based policy approximation algorithm that uses a deep-reinforcement based system dynamics model to identify the optimal control action to satisfy the tracking performance. The RL-based algorithm is motivated by the need to design an optimal controller that is generalised enough to be able to perform in various fault scenarios, without needing to be retrained or retuned. The proposed controller is trained and validated using dynamic simulations on a co-simulation platform. The platform hosts a modified version of the IEEE 123-node test system with low-inertia microgrids within the GridLAB-D™ simulation environment [31] with the RL-based controllers hosted on Python and communicating through the HELICS [32] middleware. The performance of the proposed controller is also compared to that of the previous PI controller. For a fair comparison, rather than manually tuning the PI controller, a neural-network-based tuner selects the respective controller gains.

The rest of this paper is organised as follows. Section 2 explains the concept of CVR and how it is leveraged to modify existing generator controls for improved frequency response, and Section 3 describes the RL-based controller and its design, training, and testing processes. Section 4 describes dynamic simulations performed to demonstrate the novelty of the work, and conclusions are drawn in Section 5.

2 CVR and modified AVR

CVR is a reduction in voltage to reduce active and reactive power consumption at end-use loads; it also reduces system losses to a lesser extent [33–36]. Historically, CVR has used persistent voltage reduction to reduce the annual energy consumption and/or the peak demand of distribution circuits. CVR-based systems have been in operation at the distribution level for over 30 years and there is extensive literature on the topic.

In this section, we describe leveraging CVR by engaging the A VRs of diesel generators to adjust their terminal voltages during a transient. Typically, an AVR compares a sensed voltage to a reference voltage and adjusts the field excitation of the generators to drive the difference to zero [37]. In the prior work [10], the AVR was supplemented with a manually tuned PI controller to compare the frequency as measured at its terminals with a nominal reference value (60 Hz in North America) and adjust the reference voltage as needed. Specifically, when the frequency decreased below the nominal frequency, the terminal voltage was reduced in an attempt to reduce system load, as in the CVR effect described above. The additional controller is referred to as a CVR controller. The prior work conducted simulation studies, in which one diesel generator in a networked microgrid system was equipped with the CVR controller and the primary frequency response was observed during switching transients. The results illustrated tremendous potential in the application, but additional challenges remain before CVR controllers can be deployed in the field. First, manually tuning PI controllers is extremely challenging, and the effort increases exponentially with multiple field deployments of these PI controllers in a connected system without communications. Second, distribution systems change significantly over time with respect to their topology, demand, and DERs. The CVR controller should be able to adapt to these changes and still maintain a favourable response within system constraints without requiring periodic retuning. Third, the fixed strategy approximation of RL controller also limits its adaptability to varying system topology. Finally, primary frequency response should take place on the time scale of 1–2 s, and PI controllers themselves respond sluggishly to sudden disturbances.

These drawbacks can be addressed through a neural-network-based CVR controller that can be trained before deployment through power system simulations and then set to automatically adjust with changing system conditions. This not only avoids the manual tuning process required by PI controllers but also adapts to changing system dynamics and to having multiple such controllers in the system.

The new modified AVR is illustrated in Fig. 1. The inputs to the CVR controller, in this case, are not only the reference and measured system frequencies, but also the terminal voltage, current output, and real and reactive power outputs of the generator. The objective of the CVR controller is to maintain the measured frequency, fREF, as close as possible to the reference frequency, fREF. This is done through injecting temporary adjustments of ΔVREF to the reference terminal voltage VREF into the existing exciter (SEXS – simplified excitation system) AVR [13].

During a switching transient, the associated frequency drop will result in a negative input signal to the controller. Consequently, the output of the controller ΔVREF will be negative, reducing the terminal voltage. The reduced terminal voltage decreases the average system voltage, leading to a reduction in load and electrical torque on rotating machines.

3 Reinforcement learning-based CVR controller

The main contribution of this work is the development of the RL-based CVR controller as shown in Fig. 1. Some fundamentals of RL along with the training approach are covered in this section.

3.1 Preliminaries

The CVR controller, introduced in the previous section, uses an RL-based algorithm to design an optimal policy in a sequential decision-making process. The proposed controller uses action a(t) (change in reference voltage) to control the dynamical system state s(t) (system frequency) to receive a reward R(s(t)) (improved frequency response), where t indicates the current time. Control action a(t) deterministically transitions the system to the next system state s(t + 1) using a state transition law s(t + 1) = f(s(t), a(t)). The optimal action sequence (policy) π(s) maximises the expected reward given by the following equation:

\[ V^\pi(s) = \mathbb{E}[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \ldots | s_0 = s, \pi] \] (1)

The expected reward starting at some state, s0, is denoted by V^\pi(s), as it depends on the initial state and the sequence of actions performed in each state, s(t), where γ (in (1) is a penalty factor that varies between 0 and 1 (where 0 indicates favouring policies with immediate reward and 1 indicates favouring policies with high future reward).

The optimal policy selection is a Markov decision process (MDP) [38] and can be represented by (δ, s0, Pr(δ(s), γ, R)), where δ denotes a finite countable set of states [the continuous (uncountable) state space is discretised and made countable], s0 denotes the finite countable set of control actions, Pr(δ) denotes probability distribution of the next state given a current state (for

---

**Fig. 1** Block diagram of modified AVR, including RL-based CVR controller to improve primary frequency response
the case with deterministic state transition law, this probability is 1), $\gamma \in [0, 1)$ indicates a discount factor to discount rewards obtained in the future, and $R: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the designed reward function. Preference for future reward is motivated by the need to achieve a stable system state in both the near and the long-term future.

The policy $\pi$ maps the current state to the current control action, i.e. $\pi: \mathcal{S} \rightarrow \mathcal{A}$. The optimal policy $\pi^*$ maximises the total reward over a given time and can be expressed as $\pi^*(s) = \max \ V^*(s)$, where $V^*$ is given in (1). The optimal value function is the value function obtained when the optimal policy is executed. The optimal value function also satisfies the Bellman equation, a necessary condition for optimality that connects the value function at the current state to that of previous and future states, mathematically expressed as follows:

$$V(s) = R(s) + \max_{a \in A} \sum_{s'} \gamma V^*(s'),$$

(2)

where $s_0$ is the start and $T$ is the finish time of computation.

Finally, the value iteration algorithm computes the optimal value function by iteratively using (2), starting with an initial estimate of zeros (a starting value of zero indicates neither rewarded nor penalised condition, which is an unbiased starting position). Once the optimal value function is found, the optimal policy can be calculated

$$\pi^*(s) = \arg \max_{a \in A} \sum_{s'} \gamma V^*(s'),$$

(3)

### 3.2 Problem formulation

The implementation of the RL [39] starts with the definition of the MDP, with the four parameters $(s, a, \gamma,$ and $R)$. The state space $\mathcal{S}$ is discretised into $N_s$ finite data points, namely $s_i \in \mathcal{S} \triangleq \{s + \frac{i-1}{N_s}, i \in \{1, 2, \ldots, N_s\}\}$. Similarly, the control space is discretised into $N_a$ finite data points, i.e. $a_j \in \mathcal{A} \triangleq \{0, 1\} \times \{1, 2, \ldots, N_{\bar{a}}\}$. The state space and control space are of $N_{s, \bar{a}}$ where $s, \bar{a}$ are lower and upper bounds of the state space and $\bar{a}$ are lower and upper bounds of the control space. A discretised version of the control-oriented state space model is

$$s_{i, \Delta_t} \triangleq f(s_i, a_i; \theta),$$

(4)

where $\theta$ is a model parameter and $f$ is the non-linear mapping that defines the state dynamics. For this problem, there is not a closed-form expression for $f$, which motivates the use of a multilayer perceptron (MLP)-type feedforward neural network. MLP, by definition, is a feedforward, fully connected neural network that consists of multiple hidden layers along with input and output layers. For this application, the selection of MLP is motivated by the need for a simplified neural-network architecture, which benefits the subsequent RL algorithm by reducing the computation time. The input layer consists of eight neurons: $\theta \in \mathbb{R}^8$ consists of three-phase voltage magnitude and phase angle measurements at given time $t$, $s_i \in \mathbb{R}$ consists of the state (frequency) measurement at time $t$, and $a_i \in \mathbb{R}$ consists of the designed control action at time $t$. Furthermore, three hidden layers were selected with neuron configuration $(20, 10, 5)$, and the final output layer consists of one neuron, to accommodate the desired output $s_{i, \Delta_t}$, which is the state (frequency) evolution at time $t \in [\Delta_t]$. Each layer in this MLP has a ‘relu’ (rectified linear unit) activation function, where ‘relu’ is defined as the mapping $\sigma: x \rightarrow y$ where $\sigma(x) = \max (0, x)$.

The reward function $R$ is

$$R(s) \triangleq - \| 60 - s_{i, \Delta_t} \| - \lambda \| s_{i+1, \Delta_t} - s_i \|, \quad (5)$$

where the first term signifies the tracking performance (maintain frequency at 60 Hz at each discrete time) and the second term is to reduce oscillations during state transitions. $\lambda \in \mathbb{R}$ is the relative weight of the two terms in the reward function and enables setting a priority between the two objectives. For this application, $\lambda = 0.5$, which gives more importance to the tracking performance. $\| \|, \in \mathbb{R}$ indicates 1 norm.

Finally, the discount factor, $\gamma = 0.99$, setting a higher priority on a long-term reward.

### 3.3 Reinforcement learning

The system described in (4) starts at initial state $s_0 \in \mathcal{S}$ and the designed controller takes control action $a_0 \in \mathcal{A}$. This selected control action takes the system state, $s_0$, to a new state $s_t$. From this state, the controller takes the next action, $a_1$; this process continues with successive actions until the desired state is reached. The total value function over the whole operation interval is given by

$$V(s) = R(s) + \gamma R(s) + \gamma^2 R(s) + \ldots$$

(6)

The goal of the RL algorithm is to choose control actions that maximise the total value function over time. The value function defines the expected sum of discounted rewards that the controller receives upon executing a fixed policy $\pi$, starting from the initial state $s_0$ until it reaches the desired 60 Hz

$$V^*(s) = \mathbb{E}[R(s) + \gamma R(s) + \gamma^2 R(s) + \ldots, s_0 \in \mathcal{S}].$$

(7)

This expression can be further represented in terms of a Bellman equation and shown as

$$V^*(s) = R(s) + \gamma V^*(s),$$

(8)

where the first term indicates the immediate reward term and second term represents the sum of future discounted rewards while following the policy $\pi$. This Bellman equation is used for finding an optimal value function $V^*$, while $s_0$ starts from each discretised state in the discretised interval in $s$. The optimal value function can be written as

$$V^*(s) = \max_a V^*(s) = \max_a V^*(s).$$

(9)

Finally, the policy that optimises the value function for any starting state $s_i$ in the discretised interval $s_i$ is written as

$$\pi^*(s) = \arg \max_a V^*(s).$$

(10)

The next section defines the value-iteration type of algorithm for learning optimal control policy.

### 3.4 RL optimal policy algorithm

In Algorithm 1 (Fig. 2), the optimal control actions for each discretised initial state point are computed using the value iteration algorithm, and subsequently an MLP network is used to approximate the designed optimal policy. The MLP network has an input layer with one neuron, to accommodate current state information, and an output layer with one neuron to accommodate the control action designed by the optimal policy function ($\pi(s)$, output of Algorithm 1). The MLP network has one hidden layer with five neurons; all layers are fully connected with the ‘selu’ (scaled exponential linear units) activation functions [40]. Once Algorithm 1 runs for all the discretised starting states $s_0$, this MLP network can be used to train for finding the optimal control action for any given state $s_i$ by following the optimal policy $\pi^*(s)$. 

502 IET Smart Grid, 2020, Vol. 3 Iss. 4, pp. 500-507
Input: Define $R(s, a), \gamma, N_1, N_2, g, \bar{s}, \bar{a}, \bar{\bar{a}}$

Output: $\pi^*(s)$

Discretize the domain of $s \in S$ into $N_2$ steps, i.e., $s_j \in [\bar{s}, \bar{a}]$

Discretize the domain of $a \in A$ into $N_2$ steps, i.e., $a_k \in [a, \bar{a}]$

for $i \in \{1, 2, \ldots, N_1\}$ do (This loop defines cases)

for $j \in \{1, \ldots, N_2\}$ do (This loop defines all discretized states)

for $k \in \{1, \ldots, N_2\}$ do (This loop defines all allowable control actions)

$s_{j+1} = f(s_j, a_k)$

$V(s_j) = R(s_j) + \max_{a \in A} \gamma V^*(s_{j+1})$

end for

end for

for $j \in \{1, \ldots, N_2\}$ do (Evaluate optimal policy for every discretized state)

$\pi^*(s_j) = \arg \max_{a \in A} V^*(s_{j+1})$

end for

return $\pi^*(s)$

Fig. 2 Algorithm 1: optimal RL policy approximation algorithm

4 Simulation results

This section presents the results of dynamic simulations conducted using the method presented in Section 3.

A co-simulation framework is created to test the proposed RL-based CVR controller. The power system model, simulated using GridLAB-D, collects the information on measured voltage, frequency, and real and reactive output power at every generator. This information is passed on to the individual controller for each generator. The RL algorithm is implemented using Python at the individual controller for each generator. In the co-simulation environment, these controllers interact with GridLAB-D using the open-source middleware HELICS. HELICS handles the data exchange between GridLAB-D and the Python controllers and maintains time synchronisation between the individual programs of the simulation.

4.1 Test system for simulation

The IEEE 123-Node Test System is used to represent a distribution circuit with multiple microgrids [41]. The modified version of the test system has a combination of diesel generators and inverter-connected PV generation; the inverters are compliant with IEEE Standard 1547a-2014 [17] and are grid-following. The frequency ranges for IEEE Standard 1547a-2014 are shown in Table 1. With the default settings in Table 1, the inverters will trip sooner for larger transients. Specifically, for an under-frequency transient $>59.5$ Hz and $<60.5$ Hz, the inverter will stay connected for 2.0 s. However, if the frequency drops below 59.5 Hz, the inverter will trip in 0.16 s.

The modified system has three sections that can form stable, self-regulating microgrids and a fourth section that has a high penetration of PV but cannot independently form a stable microgrid. Therefore, the modified system is divided into three microgrids and a ‘region’, which can potentially be energised from one of the microgrids but cannot form a stable microgrid on its own. A one-line diagram of the modified IEEE 123-Node test system is shown in Fig. 3. Note that the switch between nodes 54 and 94 is open initially.

The DERs of the three microgrids and one region are shown in Table 2. For each generation source, Table 2 indicates the microgrid or region where it is located, which node it is connected to, the generator type and the rated apparent power of the unit.

4.2 Simulation scenarios

In this study, controllers are trained to improve the primary frequency response of networked microgrids. Training is conducted on simulated scenarios that capture the dynamics of typical networked microgrid operations, contingencies, and protection schemes.

Table 1 IEEE Std. 1547a Frequency Ranges – default settings

| Function | Frequency, Hz | Clearing time, s |
|----------|---------------|------------------|
| UF1      | <57.0         | 0.16             |
| UF2      | <59.5         | 2.00             |
| OF1      | >60.5         | 2.00             |
| OF2      | >62.0         | 0.16             |

In resiliency mode, all scenarios start with an event that disconnects the utility and splits the test feeder into portions with interconnected microgrids. Several events are designed to trigger under-frequency and over-frequency events, involving switching operations, load restoration, PV tripping, load shedding, and so on. With a limited number of simulated scenarios, the dynamics of networked microgrid operations cannot be fully represented. Nevertheless, a principle is that the set of scenarios should involve different levels of active power imbalance. The scenario set can be expanded to improve the performance of other contingencies, e.g. the failure of one or more controllers. Another principle is to use fewer scenarios, as it can be time consuming and difficult for utilities to prepare a very large set of scenarios. Thus, the proposed method can be closer to real practice. Besides, the controllers can learn continuously to improve the performance, once they are trained with simulated scenarios and deployed in the real system.

Table 3 describes all the events simulated across the scenarios and all the consequences arising from the events. Table 4 presents all the scenarios used for the training and testing of the proposed RL-based CVR controller in terms of the sequence of simulated events and the corresponding time stamps.

4.3 Selection of training and testing datasets

The simulated scenarios of Table 4 are segregated into training and testing datasets using open-loop (no control) frequency response from each of these 10 scenarios to run a $k$-means clustering algorithm. As one could infer, each of these clusters represents a similar dataset and can help us generate a large enough training dataset (ample training data yields better performance on the testing dataset). The $k$-means algorithm results in an optimal $k$ value of 3, which captures more than 95% of the data variance. The
Table 3 Simulated contingency events across scenarios

| Event ID | Description | Consequence |
|----------|-------------|-------------|
| E1       | The circuit breaker at the substation is tripped. Also, switches 18–135, 60–160, and 97–197 are opened. | The IEEE 123-node test feeder splits into three portions: MG 1, Region 4, and interconnected MG 2 and MG 3. |
| E2       | The switch 97–197 is closed. | Region 4 is re-energised. |
| E3       | PVs of Microgrids 2 and 3 (G3 and G5) increase generation. | An over-frequency event is observed. |
| E4       | Distributed generators in MG 2 and MG 3 are overloaded. | A under-frequency event is observed and system balances out eventually. |
| E5       | Switch 108–109 is tripped. | Loads on Nodes 109, 111, 112, 113, and 114 are disconnected, leading to an over-frequency condition. |
| E6       | Switch 76–86 is opened. | MG 2 and MG 3 are disconnected. Under-frequency is observed in MG 2, while over-frequency occurs in MG 3. |
| E7       | Switch 151–300 is opened. | The IEEE 123-node test feeder splits into three portions: MG 1, Region 4, and interconnected MG 2 and MG 3. Thus, the MG 3 created by E6 is larger than the MG 3 created by E1. |
| E8       | The circuit breaker at the substation is tripped. Also, Switches 18–135, 60–160, and 72–76 are opened. | Loads are disconnected, leading to an over-frequency condition. The PV in MG 3 is tripped, as the frequency is above 62 Hz for more than 0.16 s. |
| E9       | The controller is open. | The active power generation (320 MW) of the PV in MG 3 is disconnected. |
| E10      | The PV in Microgrid 3 (G5) is tripped. | The demand is balanced and the frequency drop ends |
| E11      | A control signal is sent to increase the output of PV in MG 2 (G3) by 325 MW. | When the output of PV in MG 2 is increased after GFA devices shed load, over-frequency occurs. |
| E12      | A control signal is sent to increase the output of PV in MG 2 (G3) by 325 MW. However, due to a communication delay, this signal is delivered 3 s later. | 300 MW load is shed by GFA devices. |
| E13      | Load shedding is triggered by Grid-Friendly™appliance (GFA) devices once the frequency of a node with GFAs is lower than 59.0 Hz. |

Table 4 Simulated scenarios for controller training and validation in terms of sequence of events in Table 3 and the corresponding time stamps in parentheses

| Scenario ID | Events (timestamps in s) |
|-------------|--------------------------|
| S1          | E1 (3.000), E2 (5.501), E3 (5.854), E4 (6.014) |
| S2          | E1 (3.000)               |
| S3          | E1 (3.000), E5 (5.800)   |
| S4          | E1 (3.000), E6 (5.001), E2 (5.501) |
| S5          | E1 (3.000), E7 (5.800)   |
| S6          | E8 (3.000)               |
| S7          | E8 (3.000), E7 (5.500)   |
| S8          | E8 (3.000), E7 (4.000), E9 (5.100) |
| S9          | E1 (3.000), E10 (3.500), E11 (3.600) |
| S10         | E1 (3.000), E10 (3.500), E12 (3.600), E13 (3.648) |

4.4 Performance of the RL-based controller

Before detailing the performance of the designed RL-based control framework, it is important to define a few metrics that are used hereafter to measure the performance of the proposed controller. The following metrics are standard for measuring the asymptotic performance of a closed-loop dynamical system:

- **Rise time** ($t_r$) – Rise time is defined as the time taken for the frequency to increase from 10 to 90% of the desired final value of 60 Hz.
- **Peak time** ($t_p$) – Peak time is defined as the time taken for the frequency to reach the maximum value.
- **Overshoot** ($o$) – The amount of overshoot is defined as the percentage ratio of the difference between the maximum frequency and the desired final value of 60 Hz to the desired final value of 60 Hz.
- **Settling time** ($t_s$) – Settling time is defined as the time needed for the frequency to be bounded within a tolerance of 1% of the steady-state value of 60 Hz.
- **Steady-state error** ($e_{ss}$) – Steady-state error is defined as the difference between the desired frequency of 60 Hz and the actual final frequency.

The above metrics are used to compare the performance of the proposed RL-based CVR controller to the performance of the open-loop system. Table 5 presents the results for a few scenarios from both the training and testing datasets chosen to best display the advantages of the proposed controller. Scenarios 1, 2, and 3 are training scenarios, and Scenarios 5 and 9 are testing scenarios. The proposed controller performs significantly better in all performance metrics categories. In all scenarios, the RL control significantly improves the rise time, which means frequency at the generator terminals bounces back to 60 Hz much faster after an event than without the CVR controller. For Scenario 5, the rise time with no control is infinity because the system instability causes the violation of IEEE Std. 1547 and inverters trip offline and frequency does not stabilise. With the RL-based CVR controller, however, the inverter trip-off is avoided and frequency stabilises in a little over 1 s. Also, it is worth mentioning that the frequency overshoots after under-frequency events are reduced or completely mitigated in all simulation scenarios with the proposed RL-based CVR controller.

Figs. 4–8 show performance comparisons between no control and RL-based CVR control for a few selected simulation scenarios. Each figure depicts both a generator terminal frequency profile and the three-phase voltage profiles for both the cases. All scenarios depicted start with the tripping of the substation circuit breaker causing an under-frequency event.
In Scenario 9, shown in Fig. 4, this is followed by a PV trip in Microgrid 3 worsening the under-frequency dynamics and coming close to an inverter trip (inverters trip off if the frequency is below 59.5 Hz for 2 s or more; see Table 1) without the proposed CVR controller. At 3.6 s, the PV in Microgrid 2 increases its output. With the proposed RL-based CVR controller, however, the frequency does not drop below 59.5 Hz for more than 0.1 s at a time and the voltage at the terminal of generator G4 is also maintained close to the desired value. Though frequency drops below 59.5 Hz, the system recovers fast enough to avoid inverter trip-off due to IEEE Std. 1547. The large voltage deviation is also avoided as the CVR controller triggers a drop in terminal voltage at generator G4 temporarily reducing the load on the system till the under-frequency event is stabilised.

In Scenario 10, shown in Fig. 5, the system contains grid friendly appliance controllers [12] that stop operations when the frequency drops below 59 Hz. In the scenario, without CVR controllers, a communication delay causes the GFAs to be tripped as the PV Microgrid 2 (G3) output is not increased in time to avoid this. Later, when the PV in Microgrid 2 increases the output, an over-frequency event is caused at 6.6 s. On the other hand, the RL-based CVR controllers greatly improve the frequency response,
An over-frequency event is observed immediately followed by an under-frequency event. The RL-based CVR controller does not improve the frequency response during the initial under-frequency response when the distribution system is split. However, the next two events are completely avoided and frequency is maintained at 60 Hz when generation in the microgrids are adjusted.

In Scenario 5, shown in Fig. 7, after the distribution system is split at 3 s, the frequency drops, but PVs are not tripped. At 5.8 s, switch 151–300 is opened, separating Microgrids 2 and 3 leading to a drastic frequency drop causing inverter trip-off and system collapse. Alternatively, with the RL-based CVR controllers, the frequency response at 3 s is greatly improved and at 5.8 s the system collapse is completely avoided. Though frequency drops below 59.5 Hz, the system recovers fast enough to avoid inverter trip-off due to IEEE Std. 1547. The large voltage deviation is also avoided as the CVR controller triggers a drop in terminal voltage at generator G4 temporarily reducing the load on the system till the under-frequency event is stabilised.

In Scenario 6, shown in Fig. 8, the under-frequency events caused by the distribution system split cause highly oscillatory dynamics and the system is unable to recover in time to avoid the inverter trip-off due to IEEE Std. 1547. The RL-based CVR controller reduces the oscillations and helps the frequency to bounce back to 60 Hz in time to avoid the inverter trip-off.

All the above-presented scenarios show that the proposed RL-based CVR controller can greatly improve primary frequency response during a variety of networked microgrid operations. The controller responds well in simulation scenarios both with single-fault occurrences (such as Fig. 4) and in simulation scenarios with multiple episodes of failure (such as Figs. 5–7). The evolution of value function \( V^\pi \) with each iteration of Algorithm 1 in the design of the RL-based CVR controller is shown in Fig. 9. The reward term converges with an increasing number of runs, \( N_i \), of Algorithm 1, and indicates the existence of locally optimal policy \( \pi^* \). The desirable convergence of the reward term means there is a finite amount control inputs that enable the controller to provide primary frequency response.

4.5 Performance comparison with PI controller

This section compares the performance of the proposed RL-based controller with prior work [10]. The prior work uses a PI controller to alter the AVR voltage reference values during a transient for improved frequency response. For the comparison, a PI controller with the following architecture is tuned:

\[
Pl(t)(a, b, c) = a \pi + b + \exp(-c \pi)
\]

where \( Pl: [0, \infty) \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R} \) is the PI control input, and \( a, b, c \in \mathbb{R} \) are tunable control gains. The existing RL control input is the optimal estimate and the MATLAB `cftool` command is used to fit \( Pl(t) \) for finding the best estimate on parameters \( a, b, c \).

The performance of the RL-based controller is compared to that of the PI controller for all the ten scenarios detailed in Table 4. This section presents the comparison for Scenario 9; the comparison for the other scenarios shows similar advantages. For Scenario 9, the best fitted values for the PI controller variables are \( a = 0.04321 \) (0.0423, 0.044), \( b = 240 (-1.434, 481.4) \), and \( c = 10 \), where brackets indicate the 95% confidence interval of the respective variable.

Fig. 10 shows a comparison between the performance of the RL-based CVR controller and the designed PI controller. Although the PI controller provides recovery in the closed-loop system performance, it is around 68% slower in recovery than the RL-based CVR controller. This can be deduced from the fact that PI controllers are restrictive on their functional shape, in contrast with the RL-based CVR controller, which is designed based on the optimal value function.

The computational cost for the PI controller is associated with training only two parameters. It has a constant complexity of \( \mathcal{O}(1) \). On the other hand, the proposed MLP neural network has one neuron in the input layer and five neurons in the hidden layer. It requires training of ten parameters – weight and bias pairs for five

Fig. 8 Frequency response (Hz) and three-phase voltage (V) evolution at Generator G2 for Scenario 6

Fig. 9 Evolution of the value function with each iteration of RL-based training. This plot indicates the existence of locally optimal policy \( \pi^* \), as introduced in Section 3

Fig. 10 Comparison of frequency response at Generator G4 for RL-based controller and PI controller, for Scenario 9

maintaining the frequency closer to the desired 60 Hz, completely avoiding the GFA-level load shedding, and causing no over-frequency event when the PV in Microgrid 2 increases its output. Though frequency drops below 59.5 Hz, the system recovers fast enough to avoid inverter trip-off due to IEEE Std. 1547. The large voltage deviation is also avoided as the CVR controller triggers a drop in terminal voltage at generator G4 temporarily reducing load on the system till the under-frequency event is stabilised.
neurons in the hidden layer – and has a complexity of $O(N)$ (where $N$ is number of neurons in hidden layer). The proposed method is computationally more complex than the PI controller. However, this comes with added benefits of better frequency response without the need to manual tune and re-tune controller gains.

5 Conclusion

Networked microgrid operations improve the resilience of the power distribution network beyond isolated microgrids. However, switching transients that are required for their operations can trigger system instability and collapse, especially with the recent increase in inverter-connected renewable energy sources, that do not provide the inertia required to support system stability. Oversizing generators to provide system inertia can help, but it greatly increases capital and operation and maintenance costs. As an alternative, this work proposes leveraging traditional CVR to improve primary frequency response during networked microgrid operations. The proposed RL-based CVR controller modifies the voltage reference value fed to the standard generator controls so the load is temporarily modified during transients to increase dynamic stability and prevent system collapse. The controller uses a reinforcement-learning-based algorithm to track frequency at the desired 60 Hz. The controller greatly improves the primary frequency response, and its performance is better than that of traditional PI controllers. A reinforcement-learning-based controller also facilitates continuous adaptation to changing system conditions. Future work will explore any other neural networks that could be used without hindering the advantages already presented. The authors also intend to test the proposed CVR controller in the field and analyse the system dynamics with varying levels of CVR controller penetration.

6 References

[1] Office of Electricity Delivery and Energy Reliability.: ‘Multi-year program plan: 2017–2021’, US Department of Energy, 2017
[2] Lasseter, R.H.: ‘Microgrids’. 2002 IEEE Power Engineering Society Winter Meeting. Conf. Proc., New York, NY, USA, 2002, vol. 1, pp. 305–308
[3] Lamonica, M.L.: ‘Microgrids keep power flowing through sandy-outages’ (MIT Technology Review, USA, 2012, https://www.technologyreview.com/2012/11/07/254622/microgrids-keep-power-flowing-through-sandy-outages/)
[4] Wu, X., Wang, Z., Ding, T., et al.: ‘Microgrid planning considering the resilience against contingencies’, IET Gener. Transm. Distrib., 2019, 13, pp. 3534–3548
[5] Metelitsa, C.: ‘U.S. microgrid: Market drivers, analysis and forecast’, Greentech Media Research, Tech. Rep., 2017
[6] Sun, W., Ma, S., Alvarez-Fernandez, I., et al.: ‘Optimal self-healing strategy for microgrid islanding’, IET Smart Grid, 2018, 1, (4), pp. 143–150
[7] Moghavelli, F., Taher, S.A., Karimi, A., et al.: ‘Multi-objective design method for construction of multi-microgrid systems in active distribution networks’, IET Smart Grid, 2020, 3, (3), pp. 331–341
[8] Lasseter, R.H.: ‘Smart distribution: coupled microgrids’, Proc. IEEE, 2001, 89, (6), pp. 1074–1082
[9] Li, Z., Shahidehpour, M., Aminifar, F., et al.: ‘Networked microgrids for enhancing the power system resilience’, Proc. IEEE, 2017, 105, (7), pp. 1289–1310
[10] Schneider, K.P., Radhakrishnan, N., Tang, Y., et al.: ‘Improving primary frequency response to support networked microgrid operations’, IET Trans. Power Syst., 2019, 34, (1), pp. 659–667
[11] Hartay, C., Couy, C.: ‘Diversity: A new problem in feeder pickup’, Electr. Light Power, 1952, pp. 142–146
[12] Schneider, K.P., Tuffner, F.K., Elizondo, M.A., et al.: ‘Enabling resiliency operations across multiple microgrids with grid friendly appliance controllers’, IEEE Trans. Smart Grid, 2018, 9, (5), pp. 4755–4764
[13] Kundur, P., Balu, N.J., Lauby, M.G.: ‘Power system stability and control’ (McGraw-hill, New York, 1994)
[14] Dobakhshari, A.S., Azizi, S., Ranjbar, A.M.: ‘Control of microgrids: aspects and prospects’. 2011 Int. Conf. on Networking, Sensing and Control, Delft, Netherlands, 2011, pp. 38–43
[15] Uluski, R., Kumar, J., Venkata, S.S.M., et al.: ‘Microgrid controller design, implementation, and deployment: A journey from conception to implementation at the Philadelphia navy yard’, IEEE Power Energy Mag., 2017, 15, (4), pp. 50–62
[16] Pattabiraman, D., Lasseter, R.H., Jahnus, T.M.: ‘Comparison of grid following and grid forming control for a high inverter penetration power system’. 2018 IEEE Power & Energy Society General Meeting (PESGM), Portland, OR, USA, August 2018, pp. 1–5
[17] IEEE Standards Coordinating Committee: ‘IEEE standard for interconnecting distributed resources with electric power systems’, IEEE Std1547–2003, 2009
[18] IET Smart Grid, 2020, 3, (3), pp. 331–341