Addressing Reactive Power Sharing in Parallel Inverter Islanded Microgrid Through Deep Reinforcement Learning

Oroghene Oboreh-Snapps, Sophia A. Strathman, Jonathan Saelens, Arnold Fernandes, and Jonathan W. Kimball
Missouri University of Science and Technology, Rolla, Missouri, USA
{oogdq, ss6k4, jhskrw, arnoldanthony.fernandes, kimballjw}@mst.edu

2024 IEEE Applied Power Electronics Conference and Exposition (APEC) | 979-8-3503-1664-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/APEC48139.2024.10509093

***Abstract*—Parallel inverter microgrids (MGs) present a significant challenge in the form of inverter-based distributed generators (IBDGs) connected with varying line impedances, potentially leading to substantial reactive power-sharing errors (RPSE). This paper proposes the fusion of data-driven control into the conventional virtual synchronous generator in a bid to minimize the sharing error. First, all state variables associated with each IBDG in the microgrid are sensed and used as input data for a deep reinforcement learning (DRL) agent. Next, the DRL agent, motivated by a unique reward function, is trained to satisfy two objectives: (1) Ensure the output voltage of all IBDGs in the system stays within a safe operating boundary, (2) Ensure the RPSE for the IBDGs is minimized. The trained agent is deployed in a simple IBDG microgrid and the performance is evaluated under different system disturbances and compared with the traditional control methods.**

***Index Terms*—Reactive power sharing, Inverter Based Distributed Generator, Microgrid, Virtual Synchronous Generator, Twin Delayed Deep Deterministic Policy Gradient**

I. Introduction

Renewable energy sources, such as wind and solar power, rely on power electronic interfaces to connect to the grid and are thus known as inverter-based distributed generators (IBDGs) [1], [2]. The control of Virtual Synchronous Generators (VSGs) offers a versatile interface for Inverter Based Distributed Generators (IBDGs). In grid-connected mode, the VSG control objective is to track active and reactive power command references (PQ-mode). In islanded operation, the VSG transitions into microgrid voltage and frequency regulation mode (VF-mode) [3], [4].

Islanded microgrids commonly have multiple VSGs connected in parallel. In this configuration, the objective is to distribute the total load demand in a manner consistent with the IBDGs' base rating. For active power sharing, this expectation is typically met successfully regardless of the line impedance mismatch. However, for reactive power sharing, due to line impedance mismatch, the conventional reactive-power droop control mechanism suffers from poor reactive power sharing [5].

To minimize the reactive power sharing error (RPSE) in autonomous microgrids, two solutions are proposed in literature: communication and decentralized based strategies [6]. In terms of communication based strategies, [7] presented a control strategy that minimizes the RPSE based on voltage drop estimation for each IBDG connecting line impedance. An adaptive virtual impedance strategy is presented in [8] which requires (i) an offline calculation of the virtual impedance parameters which is stored in a 2-D table and (ii) the introduction of a secondary level controller. While the effectiveness of this approach is demonstrated, the design process is hectic and complex. Other proposed communication based methods are given in [9] and [10], but they are either highly reliant on accurate mathematical modelling or complex control loops. A favored approach is often the decentralized strategy. However, decentralized strategies cause a reduction in RPSE accuracy in comparison to the communication based strategies and rely on an accurate mathematical model of the system. For example, in [11] and [12] an enhanced droop control mechanism which involves increasing the reactive power droop gain was proposed. While the RPSE was successfully reduced, the increased droop gain negatively impacts the voltage control performance.

One common denominator between the two categories discussed is the reliance on an accurate mathematical model that describes the dynamics of the MG network. Thanks to the advancements in deep reinforcement learning (DRL), the requirement for a mathematical model can be relaxed; DRL agents learn by interacting with its environment by receiving state information, taking an action and obtaining a reward [13], [14] and [15]. To this end, the main contribution of this work involves the fusion of DRL with VSG control in a bid to both minimize the RPSE and constrain the IBDGs voltage within safe boundaries. To the best of the authors knowledge, this approach has not be utilized in a manner presented in this paper. The rest of the paper is summarized as follows: section II presents the system description, section III goes into detail regarding the DRL and reward design, and section IV and V presents the results and conclusion.

II. System Description

The conventional VSG control design for grid-tied inverters is shown in Fig. 1a. Therein, two distinct control loops are shown: the active power loop (APL) and the reactive power loop (RPL). The goal for the APL is to mimic the SG swing equation such that the inverter is capable of providing virtual inertia to enhance the frequency response. Similarly, to control the flow of reactive power in the network, the RPL is designed to mimic the excitation behavior of a SG by employing the voltage droop control. To achieve both APL and RPL control,

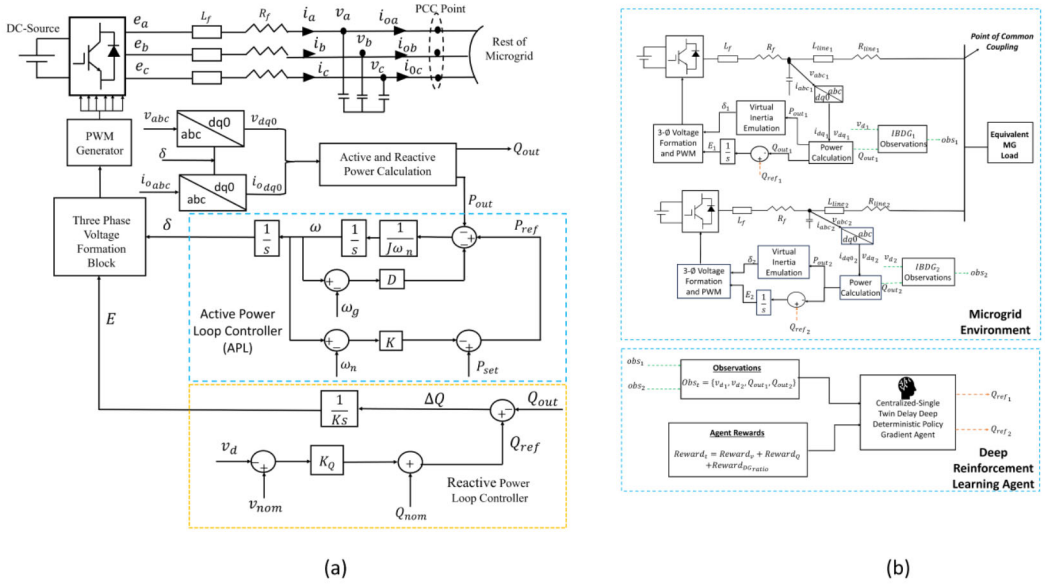

Fig. 1: Schematic of Virtual Synchronous Generator Control with (a) Conventional Reactive Power Droop Control (b) Centralized Single Agent Control for Multi-Inverter Microgrid

the output ABC current and voltage of the inverter are sensed and converted to the $dq0$ reference frame for ease of control. The output active power (P_{out}) and reactive power (Q_{out}) are calculated in the $dq0$ frame as:

$$\begin{aligned} P_{out} &= v_{od}i_{od} + v_{oq}i_{oq} \\ Q_{out} &= v_{od}i_{oq} - v_{oq}i_{od} \end{aligned} \tag{1}$$

To control the flow of active power, the VSG adopts the swing synchronous generator swing equation:

$$P_{ref} - K_p(\omega - \omega_g) - D(\omega - \omega_g) - P_{out} = J\omega\dot{\omega} \tag{2}$$

Where P_{ref}, J, D and K_p are the reference active power of the VSG, the virtual inertia, virtual damping factor and active power drooping respectively. ω and ω_g represent the speed of the virtual rotor and the reference angular speed.

From power theory, the output active power of the VSG is expressed as

$$P_{out} = \frac{3EU\sin\delta}{2X_{eq}} \tag{3}$$

With $X_{eq} = X_{line} + X_{filter}$ being the effective reactance. From (3), E is the inverter voltage magnitude which would be computed by the reactive power loop, X_{eq} is the effective reactance per phase and the load angle is represented by δ. The control output of the APL is δ which can be computed as:

$$\delta = \int (\omega - \omega_g)\, dt \tag{4}$$

Since this paper mainly focuses on the enhancing the RPL, further discussions regarding the APL are not provided. In order to control the flow of reactive power, the droop control mechanism is adopted for conventional VSG:

$$E = \frac{1}{K_i}\int \Delta Q dt \tag{5}$$

Where ΔQ is the difference between the reference reactive power Q_{ref} and Q_{out} with Q_{ref} expressed as

$$Q_{ref} = k_q(v_{nom} - v_d) + Q_{nom} \tag{6}$$

In (5)-(6), E represents the magnitude of the inverter voltage, Q_{nom} is the nominal reactive power, K_i denotes the integral control gain, and K_q represents the gain associated with the $Q-V$ droop control. The error in voltage, denoted as ΔV, is calculated by taking difference between the nominal voltage v_{nom} and actual voltage v_d and multiplying by K_q. Next, the nominal reactive power Q_{nom} is added to this result to form Q_{ref}. Subsequently, the error in the reactive power ΔQ is computed by taking the difference between Q_{ref} and Q_{out}. This error is then passed through an integral controller to generate E. Both control outputs from the RPL and APL are used to form the inverter three phase voltage and are passed to the PWM for controlling the inverter.

This work focuses on RPL as it utilizes a droop-based control which is susceptible to poor reactive power sharing. This is because the output reactive power of an IBDG depends not only on the droop gain but also on the effective impedance between the IBDG and the point of common coupling [16]. Therefore, in parallel inverter islanded MGs, when the connecting impedance is uneven, significant RPSE can be observed. To address this problem, multiple model-based solutions have been proposed. However, they often require accurate mathematical models to arrive at a good controller that minimizes the RPSE. To reduce the reliance on the model-based technique, a deep reinforcement learning (DRL) strategy can be adopted as shown in Fig.1b. DRL agents

learn by interacting with the MG environment via receiving state information, taking action, and obtaining a reward. This approach reduces the reliance on detailed mathematical modeling as the goal is for the agent to learn a policy that maximizes its reward (minimizes RPSE).

The present solution involves using a single agent for a closed MG network with mismatched impedance. The agent receives an observation set from all IBDGs in the network and generates the reactive power reference Q_{ref} for each IBDG. This allows the agent to minimize the RPSE while regulating their respective output voltages.

III. Data-Driven Control of Parallel Inverter Microgrid

The TD3 (Twin Delayed DDPG) algorithm is a reinforcement learning (RL) algorithm designed for continuous control tasks. TD3 addresses a limitation found in the critic network of the DDPG (Deep Deterministic Policy Gradient) algorithm, which tends to overestimate the value function. This overestimation can result in suboptimal policies and unstable training. To overcome this issue, TD3 incorporates several key modifications. These include; using delayed actor network updates, twin critic networks, and target policy smoothing regularization.

1) Network Architecture: The TD3 algorithm employs a network architecture illustrated in Fig. 2, comprising a total of six neural networks. This includes an actor-network parameterized by ϕ for action selection and a corresponding target actor-network parameterized by ϕ'. Additionally, the setup includes two twin critic networks parameterized by θ_1 and θ_2, responsible for Q-value estimation, along with two target twin critic networks parameterized by θ_1' and θ_2' to aid in training stability.

At the beginning of the training process, the parameters of these networks are randomly initialized. Alongside this, an empty finite buffer is created to serve as a storage cache for the agent. This buffer will be used to store and replay past experiences, facilitating the learning process of the agent.

The objective of the actor-network in TD3 is to learn the policy $\pi(s_t|a_t)$ that maximizes the expected reward. It accomplishes this by selecting the most suitable action (a_t) to take in a given state (s_t). The actor-network aims to optimize the policy to make decisions that lead to higher cumulative rewards over time, ultimately improving the performance of the agent in the RL task. The twin critic network in TD3 plays a crucial role in evaluating the action value function $Q_i(s_t, a_t|\theta_i)$. This function takes into account both the action generated by the actor network and the state information provided by the environment. By estimating the action value function, the critic network provides valuable feedback on the effectiveness and quality of the selected actions. This evaluation helps guide the learning process of the actor-network, enabling it to make more informed decisions and improve its policy over time.

To bolster the training stability in TD3, target networks are employed. These target networks are essentially frozen duplicates of the primary networks, serving as steadfast reference points throughout the training process. In the DRL, achieving convergence typically requires multiple gradient updates. Target networks play a pivotal role in mitigating the challenge of constantly shifting target values by offering consistent reference points. This stability, in turn, facilitates more effective learning, enabling the algorithm to explore a broader spectrum of actions.

The critic network consists of two paths: the state path and the action path. The state path of the critic network includes an input layer with three neurons, representing each state input. It also has a hidden layer with 64 neurons and utilizes a rectified linear unit (ReLU) activation function.

$$ReLU(x) = \max(0, x) \tag{7}$$

Both the state and action paths are then concatenated, combining their respective information. The concatenated output is subsequently passed through two additional hidden layers. The first hidden layer consists of 32 neurons, while the second hidden layer comprises 16 neurons. Both hidden layers are activated using the ReLU function. Finally, the output layer of the critic network produces a single value, corresponding to the Q-value estimation for the given state and action pair.

Conversely, the actor (policy) network takes the state input and generates an estimation of the optimal policy for the agent to follow in order to maximize the reward. The input layer of the actor network is designed to match the dimensions of the state space in the environment. Two hidden layers are employed, with the first layer consisting of 128 neurons and the second layer consisting of 64 neurons. Both hidden layers are activated using the rectified linear unit (ReLU) function. The output layer of the actor network corresponds to the dimensions of the action space, determining the actions that will be applied to the environment. In this case, the hyperbolic tangent (tanh) activation function is used in the output layer of the actor network. To ensure the predicted actions fall within the desired range, a scaling factor is applied to the output of the policy network.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{8}$$

2) Training process: As stated earlier, DRL agents learn from interacting with an environment by receiving state information, taking actions and obtaining a reward. In this work, the states(s_t) and actions(a_t) are given as;

$$s_t = [v_{d_1}, v_{d_2}, Q_{out_1}, Q_{out_2}] \tag{9}$$

$$a_t = [Q_{ref_1}, Q_{ref_2}] \tag{10}$$

The agent takes a user-defined number of training steps T_s in each episode. It has no experience on how to act in the environment at the start of training. To encourage exploration, a decaying Gaussian noise is added to the actions predicted by the actor-network as shown in (11), where ϵ is the noise and ζ is its decay factor.

$$a(t) = \pi_\phi(s_t) + \zeta\epsilon \tag{11}$$

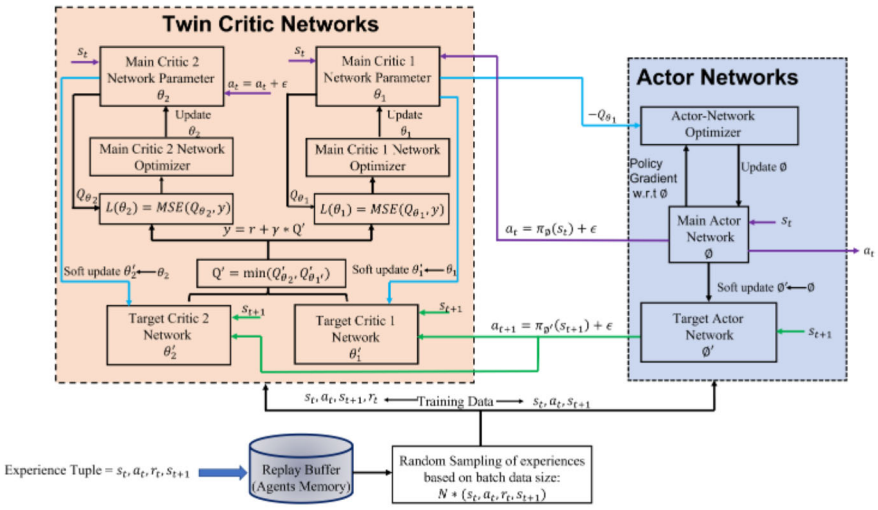

Fig. 2: TD3 Network Architecture [14]

The predicted actions a_t based on the current state s_t are applied to the environment and then the agent transitions to a new state s_{t+1}. The consequence of taking action a_t in state s_t is a reward r_t. This sequence of events, represented as s_t, a_t, r_t, s_{t+1}, forms a transition tuple that is saved in a buffer B. The experiences stored in B are randomly sampled in mini-batches and used for training the networks. As the buffer has a finite capacity, older experiences are removed to make room for newer experiences when it becomes full. This mechanism ensures that the buffer retains recent experiences, facilitating convergence during training.

3) Implementation Process: As mentioned previously, DDPG suffers from overestimation bias and sensitivity to hyper-parameter tuning. To tackle this problem, TD3 introduces the use of twin critic and target critic networks. The two critic networks receive the current state (s_t) and action (a_t) to compute their respective current action value $Q_{\theta_i}(s, a)$. In a similar manner, the target critic networks receive the next state s_{t+1} and action a_{t+1} to compute their respective next action value $Q'_{\theta_i}(s, a)$. The other two improvements introduced to address the performance of the TD3 agent are summarized below.

i) Target Policy Smoothing: As discussed in [13], deterministic policies often exhibit overestimation bias, which can lead to overly optimistic value estimates in the target network. This increased variance in the Q-values can cause issues, such as dissimilar actions producing different value estimates, potentially affecting the agent's learning process negatively. To mitigate this issue, a small amount of random noise is introduced to the target actor's actions, enhancing exploration during training. This approach encourages similar actions to produce similar value estimates, ultimately improving the learned policy of the agents.

The target actor network takes in the new state s_{t+1} and generates estimated target actions, denoted as $\tilde{a}t + 1$. Both these target actions and the new state are subsequently supplied as input to the twin target critic networks for the calculation of their respective next action values, represented as $Q'\theta_i$. To compute the target action value function, denoted as y, it is crucial to determine the minimum among the twin target critic Q-functions, as depicted in Equation 12. This approach effectively mitigates overestimation bias, which, if left unaddressed, could result in sub-optimal action value estimates.

$$y = r + \gamma \min_{i=1,2}(Q_{\theta_i'}(s', \tilde{a})) \tag{12}$$

The calculation of the target Q-value y relies on the reward r, a discount factor γ, and the minimum Q-value obtained from the target critic networks. The discount factor allows the agent to strike a balance between immediate rewards (γ=0) and long-term rewards (γ=1), influencing the agent's preference for short-term gains or long-term planning.

$$Loss_{\theta_i} = MSE(y - Q_{\theta_i}(s, a)) \tag{13}$$

To compute the loss function which is necessary to update the critic network, the mean squared error is computed individually for each critic network as shown in(13). This error is calculated by comparing the target Q-value with the Q-value predicted by each critic network. The result is a separate loss value for each critic network, which is used to update their respective parameters.

To update both the actor and critic networks, the gradients of the critic loss with respect to the weights of each critic network are calculated. These gradients are then employed to update the weights of each network using an optimizer, such as Adam.

ii) Actor Network Update: The update of the actor-network is delayed by the modulus of training step t with respect to the actor update frequency d.

$$\nabla_{\phi} J(\phi) = N^{-1} \sum \nabla_a Q_{\theta_1}(s, a)|_{a=\pi_{\phi(s)}} \nabla_{\phi} \pi_{\phi}(s) \tag{14}$$

In accordance with (14), the loss function J for the actor-network is defined with respect to its network parameters ϕ. The gradient of this loss function is computed by taking inverse of the number of training batch samples (N),

and multiply it by the summation (Σ) of the gradient of the first critic network concerning the state-action pair, i.e., $\nabla_a Q_{\theta_1}(s,a)$. This result is further multiplied by the gradient of the policy network, $\nabla_\phi \pi_\phi(s)$. In simpler terms, this process involves computing the gradient of the Q-value from the first critic network concerning the actor-network parameter ϕ. The actor loss is then determined based on the negative mean of the Q-values obtained through this procedure. The gradients of the actor loss with respect to the network parameters are subsequently computed and utilized to update the actor-network parameters.

$$\theta_i = \tau\theta_i + (1-\tau)\theta_i' \tag{15}$$

$$\phi_i = \tau\phi_i + (1-\tau)\phi_i' \tag{16}$$

Lastly, both target networks (critic and actor) are updated periodically by copying the main networks' parameters via the soft update rule with respect to a learning rate parameter τ.

TABLE I: TD3 Network Parameters

Network Parameter	Value	Network Parameter	Value
Actor Learning Rate	1×10^{-4}	Actor Network Size	[4 128 64 2]
Twin Critics Learning Rate	2×10^{-4}	Twin Critic Network Size	[4 128] State Path [2 128] Action Path [128 64 1] Common Path
Target Learning Rate	5×10^{-3}	Discount Factor γ	0.99
Buffer Length	2×10^{6}	Mini Batch Size	64
Agent Sampling Time	10ms	Action Selection Range	2500×10^{3}

TABLE II: System Parameters and Reward Penalties

System Parameter	Value	System Parameter	Value
Microgrid Capacity	6000 KVA	Inverter Nominal Voltage	13.8 kV
Filter Resistance	1.9 mΩ	Filter Inductance	0.05 mH
Microgrid Frequency	60 Hz	Virtual Inertia	3.5×10^{-5}
Virtual Damping	0.45	Penalty k_3	200
Big Penalty $[k_1,k_2,k_4]$	[5000, 5000, 500]	Small Penalty $[k_1,k_2,k_4]$	[0.05, 0.05, 0.05]
IBDG 1 Line Impedance	$R_{\text{line}} = 6m\Omega$ $L_{\text{line}} = 40.35mH$	IBDG 2 Line Impedance	$R_{\text{line}} = 0.1\Omega$ $L_{\text{line}} = 40uH$

A. Reward Function Design

The goal of any DRL agent is to find the optimal policy that maximizes the expected cumulative reward. Therefore, a good reward function must capture the problem description in order to properly guide the agent learning.

In this paper, the reward function is split into three parts:

- Voltage regulation: When IBDGs operate in autonomous mode, control of voltage and frequency becomes the priority for grid forming inverters. Therefore, the designed TD3 agent must be capable of regulating the voltage of each IBDG in the system so that the MG remains stable. In addition, it is desired that the voltage stays within a boundary of $\pm0.15pu$. Taking this into consideration, the reward for voltage regulation can be expressed as:

$$\text{Reward}_v = -k_1|e_v| \tag{17}$$

  From the above equation, $|e_v| = |V_i - V_n|$. Where, V_i represents the output voltage of the i_{th} IBDG in the system and V_n is the nominal voltage of the system all in per-unit. A penalty term k_1 is used to inform the agent on how well it is doing in terms of voltage regulation. Hence, when $|e_v|$ is greater than the desired threshold, k_1 is large; otherwise, it is set to a small penalty.

- Minimize RPS Error: As pointed out earlier, the conventional droop control suffers from poor sharing error when there exists a line impedance mismatch between both IBDGs. To address this issue, the reactive power at PCC is measured and used in designing the reward function for the agent.

$$\text{Reward}_Q = -k_2|e_q| - k_3\int(|e_q|) \tag{18}$$

  In (18), k_2 and k_3 are penalty terms associated with the RPSE. In addition, k_4 is a penalty linked with the expected contribution of the i^{th} IBDG to the rest of the IBDG in the microgrid. For the RPSE terms, $|e_q|$ is the difference between the desired output reactive power and the measured output reactive power i.e. $e_q = |Q_{share} - Q_{out}^{IBDG_1}|$ where Q_{share} is expressed as

$$Q_{\text{share}} = Q_{\text{pcc}} \cdot \frac{S_{\text{rating}}^{IBDG_i}}{S_{\text{MGcapacity}}} \tag{19}$$

  Therefore, if e_q is greater than the maximum allowable error threshold, k_2 becomes large; otherwise, it is a small penalty. On the other hand, k_3 is kept constant to motivate the agent to rapidly minimize the sharing error.

- IBDG Capacity Ratio Constraint: To guarantee that IBDGs make equitable contributions relative to their ratings in injecting reactive power, regardless of line impedance variations, the following reward term has been introduced

$$\text{Reward}_{\text{IBDGRatio}} = -k_4\text{IBDG}_{\text{ratio}} \tag{20}$$

  where the term $\text{IBDG}_{\text{ratio}}$ is expressed as:

$$\text{IBDG}_{\text{ratio}} = \frac{\text{Q}_{IBDG_{i=1}}}{\sum_{j=2}^{n}\text{Q}_{IBDG}} \tag{21}$$

  The term k_4 is the penalty associated with the $\text{IBDG}_{\text{Ratio}}$. Therefore, if the reactive power contribution of any IBDG in the network is below or above the expected contribution boundary, then k_4 is a large negative reward; otherwise k_4 is a small negative reward.

Based on (17-20), the total cumulative reward received by the agent at each time step is given as;

$$\text{Reward}_t = \text{Reward}_{v_t} + \text{Reward}_{Q_t} + \text{Reward}_{\text{IBDGRatio}} \tag{22}$$

Thus, the goal for the agent is to find the optimal control strategy that maximizes (22). Extra care should be taken when selecting the penalty values as improper selection could be detrimental to the agent's learning and performance.

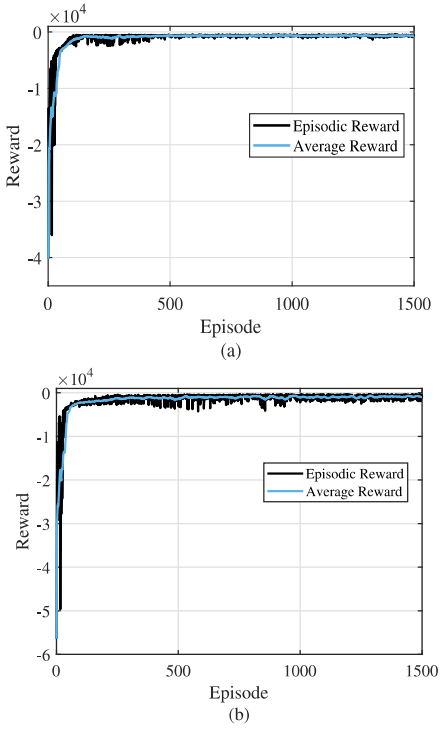

Fig. 3: Reward Curve for (a) Even IBDG Case (b) Uneven IBDG Case

IV. Results and Discussion

In order to train the TD3 agent, a 2-inverter microgrid system is designed in MATLAB/SIMULINK. The specifications for the TD3 agent and the MG are provided in Tables I and II.

Fig. 3 shows the training graph of the TD3 agent when considering even IBDGs and uneven IBDGs. The agent is trained for 1500 episodes, with each training case taking approximately 3 hours, using an ACER ASPIRE AV15-51 with 16GB RAM and a 2.90GHz processor. During this period, the agent is able to find a policy that controls the inverters in a manner that achieves the best reward.

A. Equal IBDG System

In this section, the performance of the proposed TD3 based controller is evaluated under: a) mode switching and b) load change.

1) Mode Switch: In this case study, the transition from droop based control to TD3-based control is presented. The goal is to show vividly the transient response of the control method, and provide a clear representation of the superiority of the proposed TD3 controller.

As shown in Fig 4 from 0-2 seconds, the system runs with the conventional droop based reactive power controller after which the proposed TD3 controller is activated. As expected, the droop based controller keeps the voltage at each IBDG close to 1 p.u as shown in Fig 4a. However, due to the feeder impedance mismatch, the droop controller suffers from significant reactive power mismatch as shown in Fig. 4b and

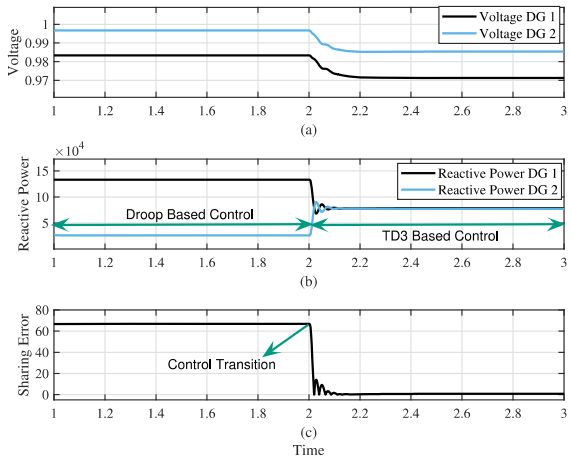

Fig. 4: Response to Control Mode Switch (a) Voltage Response (b) Reactive Power Response (c) Percentage Sharing Error

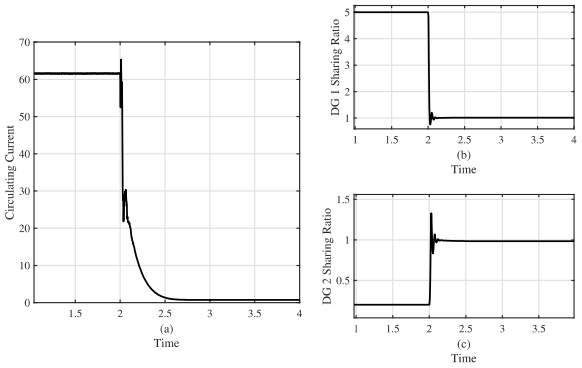

Fig. 5: Results with Equal capacity IBDG for (a) Circulating Current (b) IBDG 1 Capacity Constrained (c) IBDG 2 Capacity Constrained

4c. After 2 seconds, the TD3 based controller is activated. Recall that part of the reward presets is to ensure that the IBDG voltages remain within a $\pm$ 0.15pu band-limit. This criteria is satisfied as shown in Fig.4a when TD3 control is activated. However, unlike the droop controller, the TD3 control is capable of also sharing the reactive power according to the IBDG ratings which minimizes the sharing error as shown in Fig 4b and 4c. An added advantage of minimizing RPSE is the reduction of circulating current in the MG. As shown in Fig. 5a, by utilizing the TD3 control, the circulating current is significantly reduced. To further highlight the superiority of the proposed TD3 control, the capacity ratio of one IBDG to the other IBDG in the MG is shown in Fig. 5b and Fig 5c.

2) Load Change: In this case, the load response of the proposed TD3 control is evaluated and compared to the conventional reactive power droop control. Fig. 6a-6d shows a comparative analysis for both the reactive power droop control and TD3 based control. As illustrated, the reactive power load demand in the MG is increased and decreased at four and

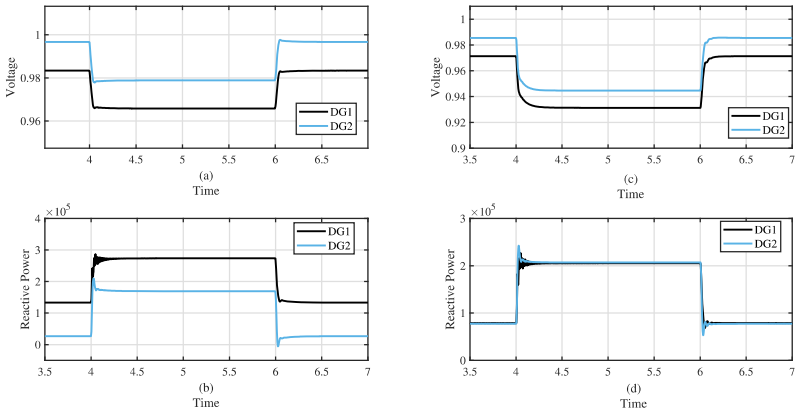

Fig. 6: Response to Load Change (a) Voltage Response with Droop (b) Reactive Power Response with Droop (c) Voltage Response with TD3 (d) Reactive Power Response with DRL

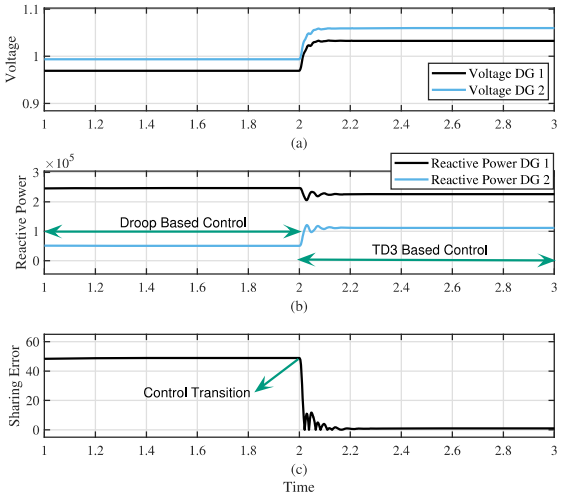

Fig. 7: Unequal IBDG Response to Control Mode Switch (a) Voltage Response (b) Reactive Power Response (c) Percentage Sharing Error

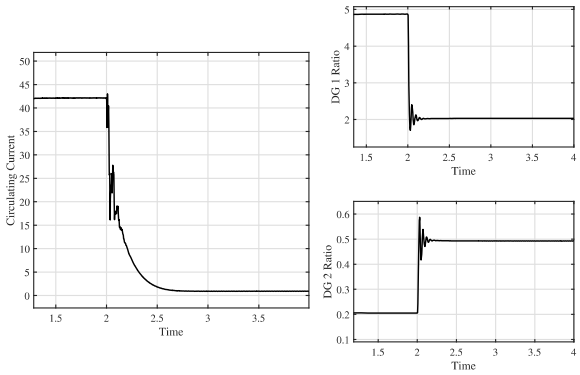

Fig. 8: Results with Unequal capacity IBDG for (a) Circulating Current (b) IBDG 1 Capacity Constrained (c) IBDG 2 Capacity Constrained

six seconds respectively. While the conventional droop-based control is capable of controlling the voltage, it is less effective at sharing the reactive power between both IBDGs as indicated in Fig.6a and 6b. However, when the TD3 controller is utilized, the voltage is kept within the desired boundary ($\pm 0.15pu$) while also ensuring the reactive power is shared evenly.

B. Unequal IBDG System

To further evaluate the performance of the proposed TD3 control, the agent is trained for two IBDGs with an unequal rating of 2:1. Again, validate the proposed TD3 control, two case scenarios are presented.

1) Mode Switch: Similar to the analysis shown with the even IBDG case, the performance of the TD3 control is compared with the reactive power droop method during a control transition event. The reactive power droop control is active from 0-2 seconds after which the TD3 based control is activated. As illustrated in Fig 7a, both the droop based and TD3 based

control operate the IBDG voltages within the $\pm 0.15pu$ bandwidth. Although, the droop controller struggles with sharing the reactive power evenly as shown in Fig 7b and 7c. In contrast, when the TD3 control is used, the reactive power is shared with respect to the IBDGs rating thus reducing the sharing error as shown in Fig. 7b and 7c. More-so, as a consequence of minimizing RPSE, the circulating current is also reduced when the TD3 control is utilized as shown in Fig. 8a.

Furthermore, Fig. 8b and 8c shows that the proposed TD3 based control restores the accurate sharing of the reactive power according to the IBDG ratings.

2) Load Change Uneven IBDG: In this case, the performance of the trained TD3 agent is evaluated under load change disturbance. Fig. 9a-9d illustrate a comparative analysis between the droop and TD3 based control with load

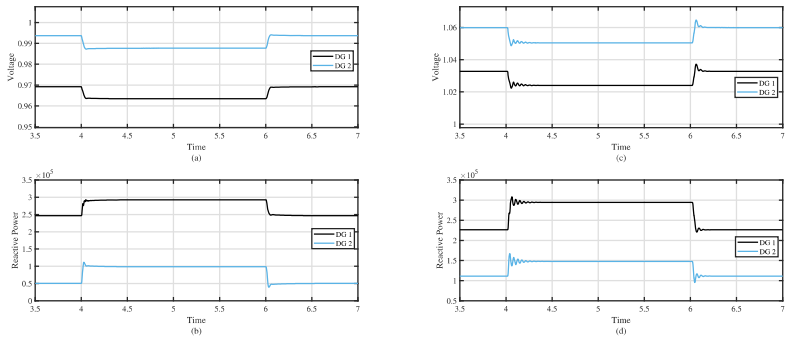

Fig. 9: Response to Load Change (a) Voltage Response with Droop (b) Reactive Power Response with Droop (c) Voltage Response with TD3 (d) Reactive Power Response with DRL

changes occurring at four and six seconds. As shown in Fig. 9a and Fig.9c, both control methods maintain the IBDGs output voltage within the desired boundary of $\pm 0.15pu$. This is important for the TD3 agent as it is one of the metrics for evaluating its performance in accordance to the reward preset on voltage regulation. However, due to feeder impedance mismatch, the reactive power delivered to the load during its increase or decrease is not proportional to the IBDG capacity. This results in significant RPSE as shown in Fig. 9b. When the trained TD3 agent is applied, the output reactive power for each IBDG corresponds to the IBDG rating, reducing the RPSE as shown in Fig. 9d.

V. Conclusion

This paper presents the fusion of deep reinforcement learning with with conventional VSG control in a bid to minimize the reactive power sharing error while ensuring voltage stability. The proposed TD3 control is achieved by receiving measurements of the voltages for both IBDGs and their output reactive power values. The reward is then designed to achieve two key functions: (i) keep the IBDG output voltage within safe bounds and (ii) minimize the RPSE. The proposed TD3-based control is compared with the conventional droop based reactive power control. Therein, the superiority of the proposed control is demonstrated for both equal and unequal IBDG cases while considering load change disturbance. Based on the results, the single agent approach reduces the percentage RPSE to $< 1\%$.

In the future, this work could be extended to include a multi-agent architecture which implies that the centralized approach presented in this paper can be modified to a decentralized approach.

References

[1] Oroghene Oboreh-Snapps, Rui Bo, Buxin She, Fangxing Fran Li, and Hantao Cui. Improving virtual synchronous generator control in microgrids using fuzzy logic control. In *2022 IEEE/IAS Industrial and Commercial Power System Asia (I&CPS Asia)*, pages 433–438. IEEE, 2022.

[2] Waqas Ur Rehman, Amirhossein Moeini, Oroghene Oboreh-Snapps, Rui Bo, and Jonathan Kimball. Deadband voltage control and power buffering for extreme fast charging station. In *2021 IEEE Madrid PowerTech*, pages 1–6. IEEE, 2021.

[3] Hassan Bevrani, Toshifumi Ise, and Yushi Miura. Virtual synchronous generators: A survey and new perspectives. *International Journal of Electrical Power & Energy Systems*, 54:244–254, 2014.

[4] OO Mohammed, AO Otuoze, S Salisu, O Ibrahim, and NA Rufa'i. Virtual synchronous generator: an overview. *Nigerian Journal of Technology*, 38(1):153–164, 2019.

[5] A Rosini, A Labella, A Bonfiglio, R Procopio, and Josep M Guerrero. A review of reactive power sharing control techniques for islanded microgrids. *Renewable and Sustainable Energy Reviews*, 141:110745, 2021.

[6] Aazim Rasool, Shah Fahad, Xiangwu Yan, Haaris Rasool, Mohsin Jamil, and Sanjeevikumar Padmanaban. Reactive power matching through virtual variable impedance for parallel virtual synchronous generator control scheme. *IEEE Systems Journal*, 17(1):1453–1464, 2023.

[7] Yun Wei Li and Ching-Nan Kao. An accurate power control strategy for power-electronics-interfaced distributed generation units operating in a low-voltage multibus microgrid. *IEEE Transactions on Power Electronics*, 24(12):2977–2988, 2009.

[8] Xiaodong Liang, Chowdhury Andalib-Bin-Karim, Weixing Li, Massimo Mitolo, and Md Nasmus Sakib Khan Shabbir. Adaptive virtual impedance-based reactive power sharing in virtual synchronous generator controlled microgrids. *IEEE Transactions on Industry Applications*, 57(1):46–60, 2020.

[9] Qing-Chang Zhong. Robust droop controller for accurate proportional load sharing among inverters operated in parallel. *IEEE Transactions on industrial Electronics*, 60(4):1281–1290, 2011.

[10] Hisham Mahmood, Dennis Michaelson, and Jin Jiang. Accurate reactive power sharing in an islanded microgrid using adaptive virtual impedances. *IEEE Transactions on Power Electronics*, 30(3):1605–1617, 2014.

[11] Nagaraju Pogaku, Milan Prodanovic, and Timothy C Green. Modeling, analysis and testing of autonomous operation of an inverter-based microgrid. *IEEE Transactions on power electronics*, 22(2):613–625, 2007.

[12] Charles K Sao and Peter W Lehn. Autonomous load sharing of voltage source converters. *IEEE Transactions on Power Delivery*, 20(2):1009–1016, 2005.

[13] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

[14] Oroghene Oboreh-Snapps, Buxin She, Shah Fahad, Haotian Chen, Jonathan Kimball, Fangxing Li, Hantao Cui, and Rui Bo. Virtual synchronous generator control using twin delayed deep deterministic policy gradient method. *IEEE Transactions on Energy Conversion*, 2023.

[15] Buxin She, Fangxing Li, Hantao Cui, Hang Shuai, Oroghene Oboreh-Snapps, Rui Bo, Nattapat Praisuwanna, Jingxin Wang, and Leon M Tolbert. Inverter pq control with trajectory tracking capability for microgrids based on physics-informed reinforcement learning. *IEEE Transactions on Smart Grid*, 2023.

[16] Jinwei He and Yun Wei Li. An enhanced microgrid load demand sharing strategy. *IEEE Transactions on Power Electronics*, 27(9):3984–3995, 2012.