IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 January 2024, accepted 17 February 2024, date of publication 18 March 2024, date of current version 28 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3379146

RESEARCH ARTICLE

An Intelligent Parameter Identification Method of DFIG Systems Using Hybrid Particle Swarm Optimization and Reinforcement Learning

XUANCHEN XIANG[1], RUISHENG DIAO[2], (Senior Member, IEEE), SHONDA BERNADIN[1], (Member, IEEE), SIMON Y. FOO[1], FANGYUAN SUN[2], (Student Member, IEEE), AND AYODEJI S. OGUNDANA[1], (Graduate Student Member, IEEE)

[1]FAMU-FSU College of Engineering, Florida State University, Tallahassee, FL 32304, USA
[2]The Zhejiang University-University of Illinois Urbana–Champaign Institute, Zhejiang University, Haining, Zhejiang 314400, China

Corresponding author: Ruisheng Diao (ruishengdiao@intl.zju.edu.cn)

ABSTRACT Precise modeling of power systems is vital to ensure stability, reliability, and secure operations. In power industrial settings, model parameters can become skewed over time due to prolonged device usage or modifications made to the control systems. Doubly-Fed Induction Generator (DFIG), one of the most prevalent generators in wind farms, is sensitive to transient occurrences. Consequently, parameter calibration of DFIG becomes a crucial focal point in power system planning and operational studies. In this paper, two baseline approaches are first developed to identify the potentially harmful parameters of the DFIG system, including the Particle Swarm Optimization (PSO) method and the state-of-the-art off-policy Reinforcement Learning (RL) method, Soft Actor-Critic (SAC). The outcomes demonstrated that the SAC method outperformed PSO, resulting in an impressive reduction of 74.67% Mean Squared Error (MSE) and a more efficient testing period. In further exploration, a novel hybrid approach called SAC-PSO is developed, with SAC being the teacher of PSO to tackle scenarios with multiple potential solutions. The results exhibited an even greater enhancement over using SAC alone, leading to a remarkable reduction of 87.84% MSE during the testing phase. The proposed method can also effectively apply to a power plant incorporating multiple wind generators.

INDEX TERMS Deep reinforcement learning (DRL), doubly-fed induction generator (DFIG), hybrid SAC-PSO, parameter calibration, parameter identification, particle swarm optimization (PSO), soft actor-critic (SAC).

I. INTRODUCTION

Over the past decades, the global power sector has observed a remarkably increased penetration of renewable generation integrated into modern power systems, among which the Doubly-Fed Induction Generator (DFIG) has garnered notable interest owing to its compact power converters, cost-efficient nature, improved quality, and distinct benefits such as decoupled active and reactive power control [1]. The main difference from the traditional system is that it employs a back-to-back converter, as shown in Fig. 1. The DFIG stator is synchronized with the grid, whereas the rotor is coupled to the converter. The back-to-back converter can supply the rotor with currents of varying frequencies. The induction generator has the flexibility to import and export reactive power independent of the turning speed, enabling various ranges of speeds of wind turbines, thus providing high stability during severe disturbances [2]. The DFIG converter has lower costs and higher efficiency than other variable speed solutions. Its interesting structure and high usability make it a dominating solution in large-scale wind farms.

The associate editor coordinating the review of this manuscript and approving it for publication was Xueguang Zhang.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

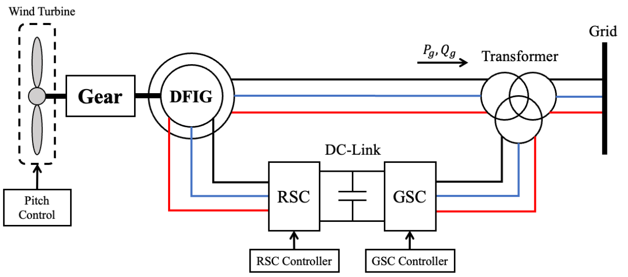


FIGURE 1. DFIG system with a back-to-back converter.

Nevertheless, DFIG has exhibited heightened susceptibility during transient occurrences [3]. In the context of power system operations, the prolonged utilization of devices frequently leads to inaccuracies in model parameters, consequently amplifying the risks associated with transient events. This makes parameter identification and calibration more essential in DFIG systems. To resolve this issue, this paper presents a novel hybrid method that takes advantage of both PSO and reinforcement learning algorithms to derive optimal model parameters that best match measurements captured during transients. The following sections of this paper are structured as follows:

Section II conducts a detailed literature survey discussing various methods for model parameter identification. In Section III, we explore the design of the parameter calibration problem and provide the mathematical models of DFIG for constructing the training environment of RL. Additionally, we introduce the principles underlying the off-policy RL algorithm known as Soft Actor-Critic (SAC) while illustrating the proposed methods. Section IV presents the research findings and offers an in-depth analysis. The performance of the two benchmark algorithms, Particle Swarm Optimization (PSO) and RL, are compared in the parameter identification process. Furthermore, we thoroughly discuss and analyze the outcomes of the proposed hybrid methods. Lastly, conclusions are drawn in Section V.

II. RELATED WORK

Power system modeling is acknowledged as one of the most intricate facets of power system analysis [4] and has been discussed regarding various calibration techniques. Traditional parameter calibration methods, such as measurement-based, component-based, and curve-fitting, are commonplace but often encounter challenges. These mathematical methods can prove difficult to converge or may yield multiple solutions. Recognizing the outstanding capabilities of Support Vector Machine (SVM) as a classification method, which can notably streamline data utilization in model development [5], [6]. Efforts have been made to enhance generalization and precision. In this regard, [7] applied the genetic algorithm (GA) method, which searches for optima across a broad parameter space. However, GA may exhibit slow convergence and converge towards local minima [4]. Similarly, Simulated Annealing (SA), characterized by its convergence during a gradual cooling process, has been leveraged as an approximation algorithm by [8]. PSO emerges as a straightforward algorithm for searching for optimal solutions amid noisy measurements. PSO has found application in parameter tuning for DFIG [9] and has exhibited commendable performance. One of the critical advantages of PSO is its ease of implementation, which positions it as one of the baseline methods adopted in this paper.

The extended Kalman filter method has recently gained widespread acceptance for parameter calibration [10], [11]. This recursive filter, encompassing prediction and correction steps, enables the estimation of dynamic systems from noisy measurements. Notably, [12] introduced EnKF (ensemble Kalman filter)for parameter estimation and calibration in DFIG. This method displayed robust performance across varying noise levels and initial errors. Intriguingly, [13] leveraged the Advanced EnKF method alongside online PMU measurements. Additionally, they introduced a sensitivity function to evaluate the influence of different parameters on the system. However, these methods need more sense of automatic adaptation for different models and are time-consuming for each converging iteration.

The advent of Deep Learning has prompted researchers to investigate its application in parameter calibration. For instance, [14] examined various Deep Learning methods on a complex 14-bus system, including CNN, LSTM, and GRU. PMU data are utilized to train these models to predict model parameters, thereby affirming the feasibility of Deep Learning in parameter calibration.

Reinforcement Learning (RL) revolves around an agent learning by interacting with its environment, primarily focusing on solving the Markov Decision Process (MDP) problems. SAC [15] provides a sample-efficient RL algorithm renowned for its prowess in handling complex decision-making challenges. It builds upon deep deterministic policy gradients (DDPG) but introduces a novel concept, the maximum entropy. This innovation enables SAC to pursue higher rewards but explore its environment effectively, striking a balance between trying new strategies and making optimal decisions. It is reported that the SAC algorithms perform well in tasks demanding both efficiency and adaptability.

SAC's remarkable stability and robustness during RL agent training make it stand out. Even when dealing with complex, continuous action spaces, SAC remains reliable. It has excelled in various domains, from robot control to decision-making in intricate settings. The SAC's efficiency in learning from past experiences positions it as a top choice for researchers and practitioners addressing real-world problems with RL, offering a powerful tool for optimizing decision-making in diverse applications. Further mathematical details will be explored in subsequent sections.

Deep Reinforcement Learning (DRL), capable of dynamic interaction with the environment, has been harnessed for real-time model calibration [16]. In a related study, [17] utilized SAC for co-calibration of dynamic model parameters across multiple events. PMU measurements served as "play-in" signals, validating the viability of SAC in this context.

Recognizing the robustness of RL and PSO, we conceived the idea of combining them for mutual benefits. Our literature review explored how others have merged RL and traditional math methods. The hybrid approaches [18], [19], [20], [21], [22], [23] provide insights, affirming the practicality of combining DRL and PSO.

Some studies employ DRL as a tool to enhance the convergence rate and stability of PSO. For example, DQL-PSO in [21] assesses the action reward at each step, directing the particles to execute the suitable actions in real-time. DDPG-PSO, proposed in [22], employs a customized Deep Deterministic Policy Gradients (DDPG) algorithm within each iteration of PSO, which leads to more effective particle velocity determination. Additionally, in another study [23], a hybrid path planning method, called Hybrid PPSO and HORL Algorithm (HPHA), fuses optimized RL and enhanced PSO. They used Hybrid Optimal Reinforcement Learning (HORL) to train the Pre-set PSO and efficiently find the best paths within the path evaluation system by considering parameters.

In another sense, PSO can also work to improve DRL's performance. In PG-PSO [24], Policy Gradient (PG) collaborates with PSO for parameter exploration within the policy network, leading to performance improvements. In PSO-QL [25], PSO accelerates the Q-learning update speed in multi-agent scenarios, resulting in efficient pose calculations for industrial robots. DRL-PPSO [26] introduces DRL with Parallel PSO, where agents aim to receive the global largest reward while minimizing processing time by sharing information with neighboring particles. Furthermore, RL-enhanced PSO employs two policy networks as proposed in [27]. These networks generate distinct normal distributions by offering two input options: One network uses the global best particle as the input state, while the other integrates its best particle.

These innovative hybrid methodologies represent promising steps toward harnessing the strengths of both DRL and PSO in various application domains. Indeed, none of the existing hybrid methods discussed incorporate imitation learning, which is the primary focus of this paper.

III. PROPOSED METHODOLOGY

A. PROBLEM STATEMENT

The problem of calibrating DFIG parameters is cast as an MDP, with the RL agent interacting with the simulated DFIG model, which serves as the environment. Within this framework, the RL agent employs the SAC algorithm, a powerful RL technique, to iteratively and dynamically adjust the DFIG model's parameters. This process unfolds as a continuous quest to identify the most optimal configuration.

Referring to [28], the primary objective function of SAC is to optimize the expected return while incorporating policy entropy, mathematically represented as:

$$J(\pi) = \mathbb{E}_{\pi_\phi}\left[\sum_t \gamma^t \left(r(s_t, a_t) - \alpha H(\pi(\cdot|s_t))\right)\right] \tag{1}$$

where π denotes the policy, $r(s_t, a_t)$ signifies the reward function at state s_t when taking action a_t. α is the temperature parameter that balances expected return and entropy $H(P)$. γ is the discount factor, and $\pi(\cdot|s_t)$ represents the policy's probability distribution of selecting an action at state s_t.

SAC adopts an off-policy actor-critic architecture featuring three networks: a policy function π_ϕ, a state value function V_ψ, and a soft Q-function Q_θ, while V_ψ and Q_θ are related through the policy.

By definition, the Q value is obtained by iteratively using a Bellman backup operation until convergence, formulated as:

$$Q_\theta(s_t, a_t) = r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}\sim p}[V(s_{t+1})] \tag{2}$$

where $V(s_t)$ represents the soft state value function:

$$V_\psi(s_t) = \mathbb{E}_\pi\left[Q_\theta(s_t, a_t) - \alpha \log \pi(a_t|s_t)\right] \tag{3}$$

The soft Q-function parameters are trained to minimize the following error:

$$J_Q(\theta) = \mathbb{E}_{(s_t,a_t)\sim D}\left[\frac{1}{2}\left(Q_\theta(s_t, a_t) - Q_{\bar{\theta}}(s_t, a_t)\right)^2\right] \tag{4}$$

where the target soft Q function with parameters $\bar{\theta}$ is derived by taking an exponentially moving average of the soft Q function weights, and the optimization of the Q function is accomplished with stochastic gradients $\nabla_\theta J_Q(\theta)$. Lastly, the policy parameters can be optimized by minimizing the expected KL divergence:

$$J_\pi(\phi) = \mathbb{E}_{s_t\sim D}\left[\mathbb{E}_{a_t\sim\pi_\phi}\left[\alpha \log(\pi_\phi(a_t|s_t)) - Q_\theta(a_t, s_t)\right]\right] \tag{5}$$

And the gradient $\nabla_\phi J_\pi(\phi)$ can then be approximated.

These equations define the SAC algorithm, highlighting its core components and their respective roles in reinforcement learning. Applying the DFIG parameter calibration problem, the training process can be formulated as in Fig. 2. More details of the functions involved are discussed in the following sections.

B. ENVIRONMENT: DFIG MODELING & SIMULATION

In accordance with the Western Electricity Coordinating Council (WECC) wind turbine model [29], as shown in Fig. 3. The dynamic behavior of DFIG is governed by three primary control models:

1) DFIG Model (WGNC): This model orchestrates high-voltage current control and low-voltage active current control, featuring an integral low-voltage power logic (LVPL) module.
2) Electric Control and Protection Model (WGNCE): WGNCE primarily oversees plant-level control and

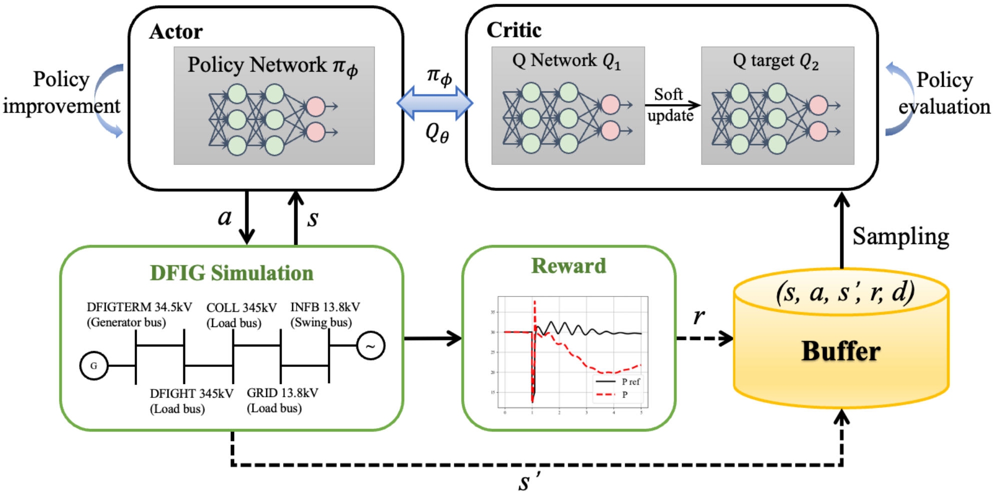


FIGURE 2. Diagram of the SAC-based method for DFIG parameter calibration.

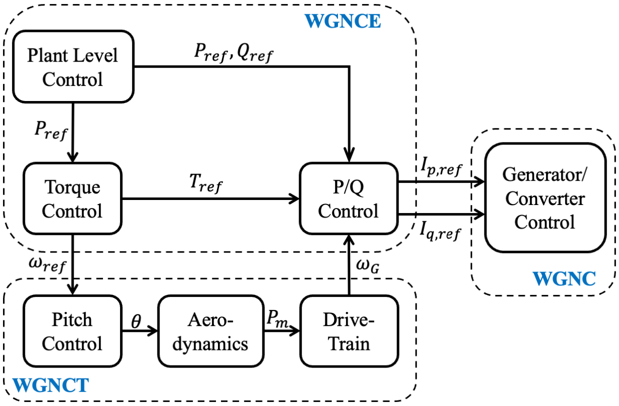


FIGURE 3. DFIG control model [29].

power quality (PQ) control, ensuring the stable operation of the wind turbine system.

3) Wind Generator Turbine and Pitch Control Model (WGNCT): WGNCT takes charge of pitch control and aerodynamic control and includes a drive-train model, which plays a pivotal role in regulating the generator's performance.

The experiment's setup is constructed utilizing the DSA Tools TSAT simulation module. It depicts a simple configuration where a DFIG bus is interconnected with the wind turbine generator, while the infinity bus signifies the swing bus, representing a point of reference. This model encompasses over 50 distinct parameters, intricately managing various aspects of the DFIG system.

The simulation process employs a dynamic file containing control parameters as its input. The total simulation duration spans 5 seconds, with a fixed time step of 8.33 milliseconds. To delve into transient phenomena, a grid fault contingency is introduced at time $T = 1$ second, providing insights into the system's response under adverse conditions.

TABLE 1. States constructed of seven parameters.

Parameter	Sen(x)	Information	Initial
X''	57.70	Sub-transient reactance	0.8
V_{lvpl2}	124.76	LVPL control high	0.9
V_{lvpl1}	41.32	LVPL control low	0.5
XI_{QMAX}	113.69	Q control	1.45
Ht_{frac}	57.11	Turbine inertia fraction	0.875
$Freq1$	131.32	Spring constant	1.8
P_{set}	467.93	Reference active power	1.0

C. STATE SPACE AND ACTION SPACE

The paramount consideration in the proposed calibration process revolves around identifying and targeting the most crucial parameters - those that wield the most significant influence over the system. To achieve this, we've harnessed a sensitivity concept introduced by [13] to extract the states of the MDP problem.

In this method, we single out one parameter for a controlled perturbation, introducing a small perturbation while keeping all other parameters constant. Subsequently, the outputs of the DFIG system are closely monitored. This meticulous process allows us to pinpoint and assess the impact of individual parameters with precision. The sensitivity function of each disturbed parameter is:

$$Sen(x) = \frac{1}{N} \sum_{k=1}^{N} |\frac{y_1 - y_2}{2 * \alpha}| \tag{6}$$

where

$$y_1 = output(x + x * \alpha)$$
$$y_2 = output(x - x * \alpha)$$

where α is the disturbing rate, we used $N = 5$ disturbing rates of [0.001, 0.01, 0.05, 0.1, 0.2]. The output metric is the average of the active power corresponding to that parameter x.

TABLE 2. SAC hyperparameter list.

hyperparameter	value
learning rate	$6e^{-3}$ linear decrease
buffer size	$10e^{6}$
batch size	256
tau	0.005
gamma	0.95
entropy coefficient	auto_0.2

Seven parameters, as shown in Table 1, turn out to have the highest calculated sensitivity value, meaning they have the highest impact on the active power. The table ranks with the vector position of each parameter. Thus, the true parameter vector is [0.8, 0.9, 0.5, 1.45, 0.875, 1.8, 1]. Drawing from extensive industrial experience, we account for a noise margin of approximately $\pm 20\%$ to encompass real-world variability. The action set is the seven adjustments corresponding to the normalized states, confined within the ± 0.2 range. After one execution, the new state is calculated as:

$$S_{t+1} = S_t + A_t$$

D. REWARD FUNCTION

Active power and reactive power are pivotal outputs in the context of power systems. Active power is the tangible output of a generator that performs useful work. Reactive power, however, is essential for maintaining voltage levels across the grid. These metrics hold paramount importance as they directly influence the stability, efficiency, and reliability of the electrical grid.

In the model parameter calibration process, the reward function quantifies how well the calibrated parameters match the actual power output. Given that power outputs exhibit distinct behaviors across different phases (before a grid fault (I), during the fault transient (II), and during recovery(III)). The reward function is developed to minimize the Euclidean distances between estimated and true power values across these phases, formulated as follows:

$$R = -\alpha_1 D(I) - \alpha_2 D(II) - \alpha_3 D(III)$$

where D represents the sum of Euclidean distances of P and Q between the reference and the estimation. It's usually appropriate to assign varying levels of importance to these phases by α's. In this case, $\alpha_1 = 1/60$, $\alpha_2 = 1/40$, and $\alpha_3 = 1/30$. The goal is considered reached when $R > -0.5$.

E. TRAINING AND BENCHMARKING

Table 2 lists the key hyperparameters employed for SAC training. These hyperparameters significantly influence the performance and behavior of the SAC agent throughout the training process. The hyperparameter tuning is not heavy, but of particular note is the entropy coefficient, set to a relatively higher initial value of 0.2. Due to its significant role in guiding the SAC algorithm, it encourages the algorithm to explore more extensively during training, aiding in escaping potential local minima and mitigating the likelihood of encountering multi-solution challenges.

Algorithm 1 PSO Algorithm Pseudocode [30]

```
Initialize particles randomly
for each iteration do
    for each particle do
        Calculate cost
        if Cost less than lowest cost $L_{best}$ in history then
            Set current cost as the new $L_{best}$
            Set current particle as the new $P_{best}$
        end if
        for each particle do
            Update particle velocity
            Update particle position
        end for
    end for
end for
```

For the benchmark setup, the PSO algorithm is adopted as a benchmark to assess and compare the performance of SAC. PSO leverages a group of particles, each representing a potential solution. These particles iteratively adjust their positions based on the local solutions they encounter, and collectively, the swarm aims to discover the global best solution. The pseudocode of PSO is presented in Algo. 1, with the cost function being defined identically to the reward function used in SAC.

Larger swarms often enhance the efficiency of PSO for more complex problems and practical applications, as demonstrated in previous studies [31]. The iteration number in PSO controls the convergence time. So, we've configured two distinct scenarios for these experiments. The first scenario involves 20 particles undergoing ten iterations, denoted as PSO20, while the second scenario features 50 particles spanning 20 iterations, denoted as PSO50.

F. THE HYBRID SAC-PSO AND PSO-SAC METHODS

To harness the strengths of SAC and PSO algorithms, we introduce two novel hybrid approaches, namely SAC-PSO and PSO-SAC, detailed in Fig. 4. These innovative methods capitalize on the concept of imitation learning, framing the algorithms as a teacher-student duo. When confronted with complex problems, the student algorithm, initially unsure of where to begin, seeks guidance from the teacher to enhance its problem-solving precision. Determining which algorithm assumes the role of the better teacher between SAC and PSO becomes the focal point of our investigation. Specifically, SAC-PSO, illustrated in Figure 4(a), positions SAC as the teacher and PSO as the student, outlining the following methodology:

The process is initiated by harnessing the predictive capabilities of the SAC algorithm. The predictions generated by SAC serve as the initial states for PSO, kickstarting its optimization journey. PSO then takes the reins and continues

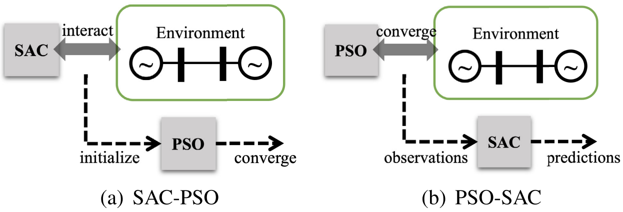


FIGURE 4. Two hybrid approaches using imitation learning.

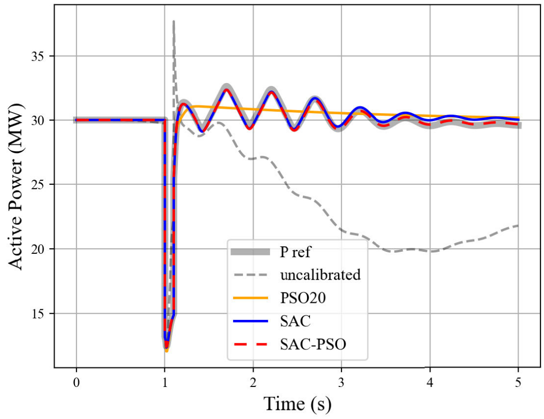


FIGURE 5. Example calibration results for the active power, comparing three algorithms.

to refine its parameters until reaching convergence. The synergy between SAC and PSO is designed to capitalize on SAC's proficiency in initial predictions and PSO's ability to fine-tune the solution further.

We have also experimented with an alternative hybrid algorithm as a comparative benchmark, as illustrated in Fig. 4(b). In this case, we reverse the roles, with PSO assuming the role of the teacher and guiding SAC. This comparative assessment offers valuable insights regarding the efficacy of SAC-PSO and its unique contributions to the field of DFIG model parameter calibration.

IV. CASE STUDIES AND ANALYSIS

A. EXAMPLE CALIBRATION CASE

Fig. 5 provides an illustration of the calibration problem. The uncalibrated curve displays substantial mismatches, underscoring the issue at hand. Notably, three algorithms—PSO20, SAC, and SAC-PSO—exhibit varying degrees of success in calibrating the curves. In this particular case, PSO20 falls short in capturing the resonance during the recovery phase, while SAC-PSO emerges as the algorithm that aligns most closely with the reference curve.

It is important to note that the performances of these algorithms may be contingent on different cases with diverse initializations. Further insights and additional results are expounded upon in subsequent sections for a comprehensive understanding.

B. SAC V.S. PSO

Fig. 6 provides the training progression of the SAC algorithm, showing the episode average reward and the actor loss during training. Notably, the episode's average reward steadily converges toward the maximum achievable reward, a noteworthy milestone achieved at approximately 15,000 steps into the training process.

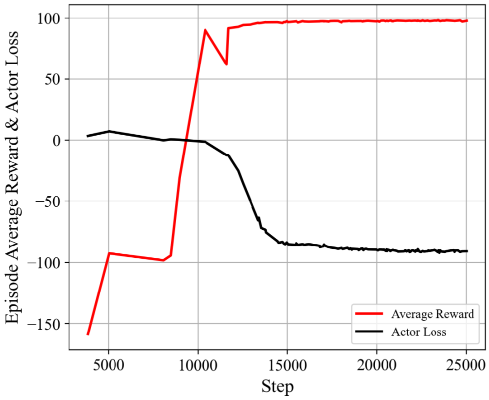


FIGURE 6. SAC training progress.

To ensure clarity in presentation, the figure displays explicitly data for the initial 25,000 training steps. The complete training process extends beyond this range. An observation is the correlation between the actor loss and the reward curves, reflecting their synchronized behavior. Furthermore, for each episode initiated with a random start, the trained SAC agent demonstrates remarkable efficiency by identifying a viable solution within a mere three steps.

To assess the algorithms' precision, we employ the Mean Squared Error (MSE) of the states as our primary metric. The results are summarized in Table 3. When examining the MSE, we observe the following insights: Firstly, both SAC and PSO algorithms effectively reduce MSE after training. SAC, for instance, achieves a substantial improvement, reducing the MSE from 0.0150 to 0.0083, marking an approximate 76.67% decrease. PSO20 and PSO50 demonstrate the effectiveness by achieving a significant reduction of 29.19% and 60.49%, respectively. Comparing PSO20 and PSO50, PSO50 outperforms PSO20. This is attributed to PSO50's ability to explore a wider candidate set and perform more iterations. Regarding accuracy across the three methods, SAC emerges as the top performer.

Moving on to the time aspect, the training process for SAC is indeed time-consuming, but it's a one-time job. Subsequent testing requires only 20 seconds, making it highly efficient in real-time applications. In contrast, PSO's calibration time varies, especially when the initialization is less accurate, demanding more iterations. Furthermore, PSO's performance heavily relies on the number of particles and iterations. As we can observe, PSO50 takes considerably more time than PSO20. This suggests a trade-off between time and accuracy in the case of PSO.

Regarding algorithm implementation, PSO is relatively straightforward and doesn't necessitate extensive parameter tuning. For SAC, some hyperparameter tuning is advisable to prevent stagnation and ensure effective learning. Crafting an

TABLE 3. Summary of algorithms' performance in parameter calibration.

Algo.	start MSE	mid MSE	end MSE	Improve (%)	Wall Train time	Wall Test time (s)
SAC	0.0150	-	0.0038	**74.67**	1 Day (CPU)	20 (per step)
PSO20	0.0161	-	0.0114	29.19	-	540 (per test)
PSO50	0.0162	-	0.0062	60.49	-	2675 (per test)
SAC-PSO	0.0148	0.0025	0.0018	**87.84**	-	2733 (per test)
PSO-SAC	0.0164	0.0048	0.0022	86.49	-	2732 (per test)

TABLE 4. States constructed of eleven parameters.

Parameter	Sen(x)	Information	Initial
X''	57.70	Sub-transient reactance	0.8
V_{lvpl2}	124.76	LVPL control high	0.9
V_{lvpl1}	41.32	LVPL control low	0.5
P_{MAX}	26.91	Max power reference	1.12
P_{wrat}	21.17	Ramp rate	0.45
$WP100$	32.57	Generator speed level	1.2
XI_{QMAX}	113.69	Q control	1.45
Ht_{frac}	57.11	Turbine inertia fraction	0.875
$Freq1$	131.32	Spring constant	1.8
D_{shaft}	17.24	Damping coefficient	1.5
P_{set}	467.93	Reference active power	1.0

appropriate reward function can be challenging but crucial, as it influences the presence of multiple optimal solutions. However, when well-constructed, SAC excels at handling scenarios with multiple solutions.

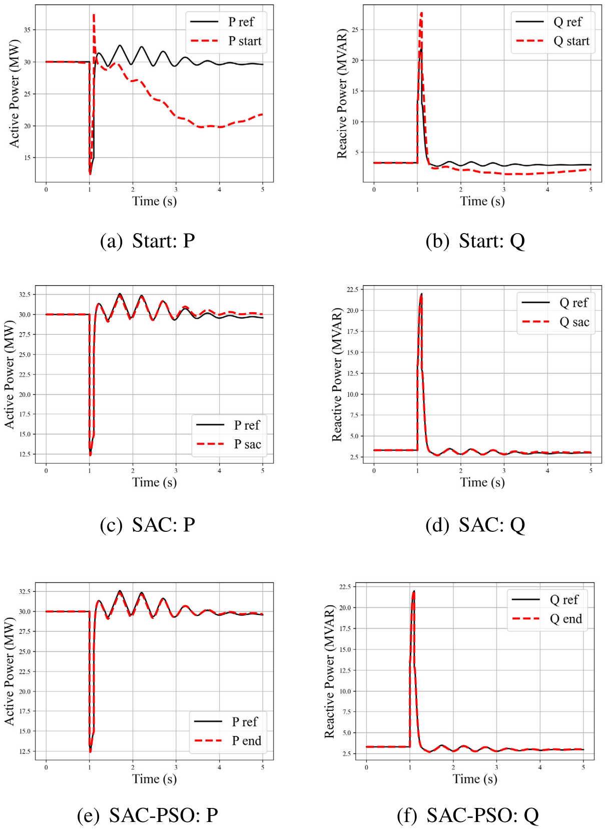
Summarizing the three methods, we can conclude that SAC, while requiring longer preparation time, offers faster calibration during testing. On the other hand, PSO is easier to implement but tends to be slower during calibration. The decision between the two options depends on specific requirements, which involve balancing time constraints and accuracy needs during calibration.

C. SAC-PSO V.S. PSO-SAC

Moving to the outcomes generated by the hybrid algorithms, Fig. 7 portrays a SAC-PSO calibration scenario. This visual representation illustrates the calibration process, highlighting instances of initial mismatches and the subsequent transformation achieved through the calibrated results.

When the calibration process commences with random initialization, as shown in the first row, the power values exhibit notable discrepancies. However, under the guidance of the teacher agent (SAC), these errors are substantially reduced, as presented in the second row. Except for a slight recovery phase mismatch, SAC already performs excellently. From the third-row figures, the student agent (PSO) takes over, further refining the states to mitigate any remaining noise. It provides evidence that the SAC-PSO hybrid approach excels in addressing the model parameter calibration challenges, leading to precise and accurate results.

The evolution of MSE, as detailed in Table 3, mirrors this progress. Beginning at 0.0148, the MSE declines to 0.0025 and ultimately reaches an impressive low of 0.0018, marking it the lowest MSE achieved among all algorithms. This remarkable improvement represents an average rate of enhancement of 87.84%.



(a) Start: P

(b) Start: Q

(c) SAC: P

(d) SAC: Q

(e) SAC-PSO: P

(f) SAC-PSO: Q

FIGURE 7. Example calibration results of SAC-PSO: (a) (b) are the mismatch of powers with bad parameters. (c) (d) are results after the calibration prediction of SAC. (e) (f) are the final calibrated results refined by PSO.

In the case of PSO-SAC, the approach leverages a similar conceptual framework and yields high improvement as well compared to individual methods, reaching 86.49%. It proves the effectiveness of hybrid techniques. Our testing shows it is not as good as SAC-PSO. It's important to note that PSO notably relies on its initialization and is easy to get stuck at local minima. In this context, the teaching phase doesn't fully capitalize on PSO's characteristics. Furthermore, SAC incorporates an end-state criterion, indicating the episode is finished when specific reward conditions are met. Consequently, when teacher PSO yields acceptable results, SAC promptly marks the episode as "Done" limiting further improvements. Conversely, when PSO performs poorly, SAC perceives little distinction between this and a random start scenario, as both routes lead toward the "Done" state rather than a "Better" state unless an extremely decent reward

TABLE 5. Summary of SAC and SAC-PSO performance with realistic dataset.

Algo.	start MSE			mid MSE			end MSE		
	States	P	Q	States	P	Q	States	P	Q
SAC	0.0160	4.3620	0.5811	-	-	-	0.0143	0.2874	0.01765
SAC-PSO	0.0168	3.9403	0.5160	0.0084	0.2742	0.0279	0.0084	0.1862	0.0088

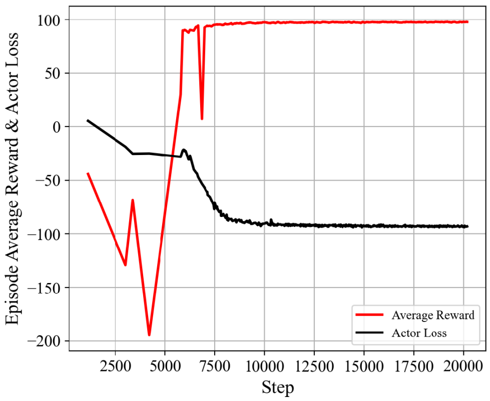


FIGURE 8. SAC training progress with realistic dataset.

function is formulated. This concept of PSO-SAC seems to be a waste of computing resources, limiting further improvement. That is the reason we focus more on SAC-PSO.

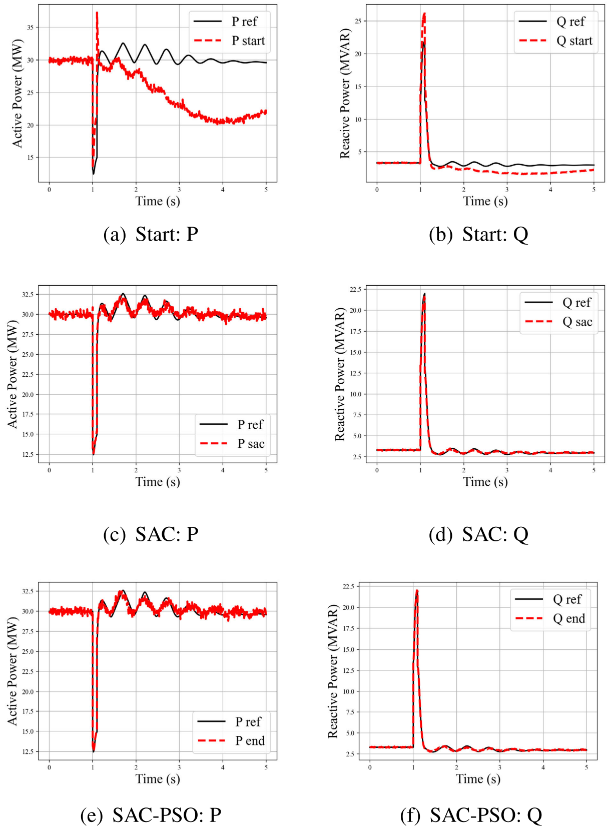
D. REALISTIC DATA ANALYSIS

The above experiments ensured the feasibility of the proposed methods. To further validate the hybrid SAC-PSO algorithm, we continue to add four more control parameters and apply noise to the power output to mimic realistic data. The new set of states is shown in Table. 4, with eleven control parameters constructing the state vector. The noisy level of the outputs is set to be 1%.

The training progress is depicted in Fig. 8. Due to the larger dataset with increased parameters and noise, the training process on a CPU took approximately three days. The potential for enhancing this process is evident through the utilization of GPUs in the future. The graphical representation illustrates clear convergence, notably around the 8,000-step mark. By the first findings presented in Table 5, the trained SAC model exhibited commendable calibration of active and reactive power, notably reducing the MSE for active power from 4.36 to 0.28. However, It is worth noting that the MSE for states did not show very high improvement, about 10%.

Using the hybrid SAC-PSO method has demonstrated significant improvements in power calibration, as evident from the outcomes illustrated in Fig. 9. The initial mismatch, particularly noticeable at the start, is effectively addressed through calibration, showcasing a substantial enhancement in power accuracy. Due to the robustness of SAC and the application of noise, it is challenging to visualize the second phase using PSO. Nevertheless, the significant improvements in power are quantified in Table 5.

However, as indicated by MSE results, mitigating state mismatches shows a more nuanced picture. While there is improvement after the application of SAC, the subsequent introduction of PSO results in little improvement in MSE for states. Several factors contribute to this observation. Firstly, the performance of these algorithms is intricately linked to the construction of the reward function in SAC and the cost function in PSO. Directly using these functions from the no-noise model may contribute to this discrepancy. Secondly, some parameters exhibit high sensitivity, while others have low sensitivity. The comprehensive MSE considers all states, leading to varying degrees of calibration success. Some parameters may be well-calibrated, while others, despite not impacting power accuracy, may contribute to the total MSE not improving or even increasing. In this context, a higher MSE might not necessarily indicate a deterioration in performance in a meaningful sense.



(a) Start: P

(b) Start: Q

(c) SAC: P

(d) SAC: Q

(e) SAC-PSO: P

(f) SAC-PSO: Q

FIGURE 9. Example Calibration results of SAC-PSO for realistic data: (a) (b) are the mismatch of powers with bad parameters. (c) (d) are results after the calibration prediction of SAC. (e) (f) are the final calibrated results refined by PSO.

In conclusion, our primary objective of calibrating performance based on power remains successfully achieved. Addressing the mismatch in states suggests the potential for improvement through modifications to the reward functions. Furthermore, embedding control parameter values directly into the rewards could be a promising avenue for enhancing efficiency in future iterations. This underscores the ongoing potential for refinement and optimization in our calibration approach.

V. CONCLUSION

A. SUMMARY

In pursuit of achieving precise parameter calibration for DFIG modeling, our research delved into the application of several algorithms, including SAC, PSO, and the proposed innovative hybrid approaches known as SAC-PSO and PSO-SAC.

We initiated our exploration by establishing the feasibility of Deep Reinforcement Learning, focusing on SAC, in the context of parameter calibration for DFIG power systems. The results demonstrated the remarkable performance of SAC, significantly outperforming the traditional mathematical method represented by PSO. This performance evaluation enabled a comprehensive analysis of the strengths and advantages offered by both algorithms, providing valuable insights for industries seeking the most suitable method based on criteria such as accuracy, efficiency, and cost-effectiveness.

Subsequently, we introduced the hybrid methodologies, SAC-PSO and PSO-SAC, which exhibited superior performance compared to the individual algorithms employed in isolation. These hybrid approaches are promising for enhancing parameter calibration processes, emphasizing their potential in real-world applications.

In our final analysis, we conducted a realistic experiment by introducing additional parameters and noise into the power data. While this posed challenges in refining the reward functions, as highlighted in the earlier analysis, the power calibration proved to be a success. Despite the heightened complexity introduced by these additional factors, the algorithm demonstrated effective utility, showcasing its robustness and efficiency in addressing real-world complexities. This underscores the algorithm's ability to handle intricate scenarios and its overall efficacy in power calibration.

B. FUTURE APPLICATION DIRECTIONS

Notably, our calibration methodology extends its utility beyond the calibration of individual power generators, finding relevance in the context of entire power grids. This expanded scope underscores its versatility and underlines the potential for significantly enhancing the performance and reliability of power grids on a larger and more impactful scale. Due to the generalized nature of our methods, we can adapt them to a wide range of generators. Experimenting with different generators not covered in this study will offer insights into their effectiveness across diverse energy systems. On a larger scale with identical generators, we can calibrate one generator's parameters and then apply them to all others. Even with different types of generators, this method could still be technically accomplished by utilizing a larger state vector containing all distinct control parameters.

Additionally, we're keen on integrating our methods with other renewable energy sources. For instance, exploring how our methods can be incorporated into solar energy systems is an exciting prospect, aiming to contribute to the broader improvement of renewable energy utilization. By exploring these future research avenues, our goal is to enhance the applicability of our methods and make meaningful contributions to the advancement of renewable energy technologies.

C. CHALLENGES AND FUTURE WORK

Several practical challenges persist in implementing these techniques in industries. It is important to note that we focus on applying the proposed technique in transient cases. Primarily, transient cases occur infrequently, making it challenging to obtain data related to these instances. Given the diversity of faults causing transient cases, acquiring such data poses a realistic hurdle for future work. Secondly, the lack of transparency in real-world infrastructure complicates matters. While striving to find simulation models that closely mimic reality, obtaining the actual industry model remains elusive, potentially leading to inaccuracies in modeling. Thirdly, neglecting to switch details can significantly impact calibration results, underscoring the importance of considering these specifics in the process.

Future efforts could pivot towards leveraging actual PMU (Phasor Measurement Unit) data to fortify these methodologies' practicality and real-world applicability. Incorporating this data source would significantly enhance model validation, elevating the credibility of these approaches. This avenue of research offers the potential for even greater precision in parameter calibration practices within power systems. Addressing the aforementioned challenges remains a priority, and our commitment lies in presenting accurate solutions to these complexities.

Additionally, within our hybrid algorithm, the two algorithms collaborate in a cooperative manner. Exploring alternative concepts, such as competitive frameworks, promises further enhancement. Embracing diverse approaches could unlock new dimensions of efficiency and effectiveness in our methodologies.

REFERENCES

[1] A. Awasthi and R. Diwan, "Study for performance comparison of SFIG and DFIG based wind turbines," *Int. J. Latest Trends Eng. Technol.*, vol. 2, no. 4, pp. 1–10, 2013.

[2] M. J. Harandi, S. G. Liasi, E. Nikravesh, and M. T. Bina, "An improved control strategy for dfig low voltage ride-through using optimal demagnetizing method," in *Proc. 10th Int. Power Electron., Drive Syst. Technol. Conf. (PEDSTC)*, 2019, pp. 464–469, doi: 10.1109/PEDSTC.2019.8697267.

[3] T. T. Phan, V. L. Nguyen, M. J. Hossain, A. N. To, H. T. Tran, and T. N. Phan, "Transient responses of the doubly-fed induction generator wind turbine under grid fault conditions," in *Proc. Int. Conf. Adv. Comput. Appl. (ACOMP)*, Nov. 2016, pp. 97–104, doi: 10.1109/ACOMP.2016.023.

[4] F. S. Chassin, E. T. Mayhorn, M. A. Elizondo, and S. Lu, "Load modeling and calibration techniques for power system studies," in *Proc. North Amer. Power Symp.*, Aug. 2011, pp. 1–7, doi: 10.1109/NAPS.2011.6024878.

[5] J. Ma, Z. Y. Dong, and P. Zhang, "Using a support vector machine (SVM) to improve generalization ability of load model parameters," in *Proc. IEEE/PES Power Syst. Conf. Expo.*, Mar. 2009, pp. 1–8, doi: 10.1109/PSCE.2009.4839969.

[6] M. Jin, H. Renmu, and D. J. Hill, "Load modeling by finding support vectors of load data from field measurements," *IEEE Trans. Power Syst.*, vol. 21, no. 2, pp. 726–735, May 2006, doi: 10.1109/TPWRS.2006.873101.

[7] S. Z. Zhu, Z. Y. Dong, K. P. Wong, and Z. H. Wang, "Power system dynamic load identification and stability," in *Proc. Int. Conf. Power Syst. Technol.*, 2000, pp. 13–18, doi: 10.1109/icpst.2000.900024.

[8] Y.-H. Song and M. R. Irving, "Optimisation techniques for electrical power systems. Part 2: Heuristic optimisation methods," *Power Eng. J.*, vol. 15, no. 3, pp. 151–160, Jun. 2001, doi: 10.1049/pe:20010307.

[9] F. Wu, P. Ju, and X. P. Zhang, "Parameter tuning for wind turbine with doubly fed induction generator using PSO," in *Proc. Asia–Pacific Power Energy Eng. Conf.*, Mar. 2010, pp. 1–4.

[10] K. Kalsi, Y. Sun, Z. Huang, P. Du, R. Diao, K. K. Anderson, Y. Li, and B. Lee, "Calibrating multi-machine power system parameters with the extended Kalman filter," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Jul. 2011, pp. 1–8, doi: 10.1109/PES.2011.6039224.

[11] Z. Huang, P. Du, D. Kosterev, and B. Yang, "Application of extended Kalman filter techniques for dynamic model parameter calibration," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Jul. 2009, pp. 1–8, doi: 10.1109/PES.2009.5275423.

[12] R. Fan, Z. Huang, S. Wang, R. Diao, and D. Meng, "Dynamic state estimation and parameter calibration of a DFIG using the ensemble Kalman filter," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Jul. 2015, pp. 1–5, doi: 10.1109/PESGM.2015.7285990.

[13] R. Huang, R. Diao, Y. Li, J. Sanchez-Gasca, Z. Huang, B. Thomas, P. Etingov, S. Kincic, S. Wang, R. Fan, G. Matthews, D. Kosterev, S. Yang, and J. Zhao, "Calibrating parameters of power system stability models using advanced ensemble Kalman filter," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 2895–2905, May 2018, doi: 10.1109/TPWRS.2017.2760163.

[14] S. Wshah, R. Shadid, Y. Wu, M. Matar, B. Xu, W. Wu, L. Lin, and R. Elmoudi, "Deep learning for model parameter calibration in power systems," in *Proc. IEEE Int. Conf. Power Syst. Technol. (POWERCON)*, Sep. 2020, pp. 1–6, doi: 10.1109/POWERCON48463.2020.9230531.

[15] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," 2018, *arXiv:1801.01290*.

[16] Y. Tian, M. A. Chao, C. Kulkarni, K. Goebel, and O. Fink, "Real-time model calibration with deep reinforcement learning," *Mech. Syst. Signal Process.*, vol. 165, Feb. 2022, Art. no. 108284.

[17] S. Wang, R. Diao, C. Xu, D. Shi, and Z. Wang, "On multi-event co-calibration of dynamic model parameters using soft actor-critic," *IEEE Trans. Power Syst.*, vol. 36, no. 1, pp. 521–524, Jan. 2021, doi: 10.1109/TPWRS.2020.3030164.

[18] D. Hein, A. Hentschel, T. A. Runkler, and S. Udluft, "Reinforcement learning with particle swarm optimization policy (PSO-P) in continuous state and action spaces," *Int. J. Swarm Intell. Res.*, vol. 7, no. 3, pp. 23–42, Jul. 2016, doi: 10.4018/ijsir.2016070102.

[19] Q. Ge, C. Guo, H. Jiang, Z. Lu, G. Yao, J. Zhang, and Q. Hua, "Industrial power load forecasting method based on reinforcement learning and PSO-LSSVM," *IEEE Trans. Cybern.*, vol. 52, no. 2, pp. 1112–1124, Feb. 2022, doi: 10.1109/TCYB.2020.2983871.

[20] S. Ding, W. Du, X. Zhao, L. Wang, and W. Jia, "A new asynchronous reinforcement learning algorithm based on improved parallel PSO," *Appl. Intell.*, vol. 49, no. 12, pp. 4211–4222, Dec. 2019, doi: 10.1007/s10489-019-01487-4.

[21] K. Elbaz, A. Zhou, and S.-L. Shen, "Deep reinforcement learning approach to optimize the driving performance of shield tunnelling machines," *Tunnelling Underground Space Technol.*, vol. 136, Jun. 2023, Art. no. 105104, doi: 10.1016/j.tust.2023.105104.

[22] J. Song, B. Zhang, and J. Lia, "Deep reinforcement learning empowered particle swarm optimization for aerial base station deployment," in *Proc. IEEE Appl. Sens. Conf. (APSCON)*, Jan. 2023, pp. 1–3, doi: 10.1109/APSCON56343.2023.10101173.

[23] X. Liu, D. Zhang, T. Zhang, J. Zhang, and J. Wang, "A new path plan method based on hybrid algorithm of reinforcement learning and particle swarm optimization," *Eng. Computations*, vol. 39, no. 3, pp. 993–1019, Mar. 2022, doi: 10.1108/ec-09-2020-0500.

[24] T. Liu, L. Li, G. Shao, X. Wu, and M. Huang, "A novel policy gradient algorithm with PSO-based parameter exploration for continuous control," *Eng. Appl. Artif. Intell.*, vol. 90, Apr. 2020, Art. no. 103525, doi: 10.1016/j.engappai.2020.103525.

[25] X. Lu, Y. Chen, and Z. Yuan, "A full freedom pose measurement method for industrial robot based on reinforcement learning algorithm," *Soft Comput.*, vol. 25, no. 20, pp. 13027–13038, Oct. 2021, doi: 10.1007/s00500-021-06190-6.

[26] A. Pradhan, S. K. Bisoy, S. Kautish, M. B. Jasser, and A. W. Mohamed, "Intelligent decision-making of load balancing using deep reinforcement learning and parallel PSO in cloud environment," *IEEE Access*, vol. 10, pp. 76939–76952, 2022, doi: 10.1109/ACCESS.2022.3192628.

[27] D. Wu and G. G. Wang, "Employing reinforcement learning to enhance particle swarm optimization methods," *Eng. Optim.*, vol. 54, no. 2, pp. 329–348, Feb. 2022, doi: 10.1080/0305215x.2020.1867120.

[28] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft actor-critic algorithms and applications," 2018, *arXiv:1812.05905*.

[29] W. Remtf. (2013). *Specification of the Second Generation Generic Models for Wind Turbine Generators*. [Online]. Available: https://www.wecc.biz/Reliability/WECC-Second-Generation-Wind-Turbine-Models-012314.pdf

[30] H.-P. Dai, D.-D. Chen, and Z.-S. Zheng, "Effects of random values for particle swarm optimization algorithm," *Algorithms*, vol. 11, no. 2, p. 23, Feb. 2018, doi: 10.3390/a11020023.

[31] A. P. Piotrowski, J. J. Napiorkowski, and A. E. Piotrowska, "Population size in particle swarm optimization," *Swarm Evol. Comput.*, vol. 58, Nov. 2020, Art. no. 100718, doi: 10.1016/j.swevo.2020.100718.

XUANCHEN XIANG received the B.S. degree in electrical engineering from Huazhong University of Science and Technology, China, in 2017, and the M.S. degree from Florida State University, in 2018. She is currently pursuing the M.S. degree with the Online Master of Science in Computer Science (OMSCS), Georgia Institute of Technology, and the Ph.D. degree with Florida State University. She is also a Dedicated Scholar. Her research interests include deep reinforcement learning control within the field of power systems and DRL algorithms and their innovative applications in machine learning.

RUISHENG DIAO (Senior Member, IEEE) received the Ph.D. degree from the Department of Electrical Engineering, Arizona State University. He is currently with The Zhejiang University-UIUC Institute, Zhejiang University, Haining, China, as a Tenured Associate Professor and the Director of Renewable Power System Simulation and Intelligent Control. His research interests include high-fidelity power system modeling, renewable integration, and applications of artificial intelligence and high-performance computations in modern power systems. He is also a fellow of IET.

SHONDA BERNADIN (Member, IEEE) received the B.S. degree in electrical engineering from Florida Agricultural and Mechanical University, in 1997, the M.S. degree in electrical engineering from the University of Florida, in 1999, and the Ph.D. degree in electrical engineering from Florida State University, in 2003. She is currently the Google Endowed Professor in computer engineering with the FAMU-FSU College of Engineering. Her research interests include digital signal processing (DSP), artificial intelligence (AI), machine learning (ML), and natural language processing (NLP). Specifically, she investigates the use of AI/ML, data analysis, DSP in applications in 3D printing, ethical generative AI, blockchain, cybersecurity, semiconductor manufacturing, and health sciences. She is very active in developing engineering education and outreach programming that seeks to broaden participation and build capacity in engineering. She was a recipient of the FAMU Research Excellence Award, in 2021, INSIGHT into Diversity Inspiring Leaders in STEM Award, in 2017, and received the Excellence in Service Award from the FAMU-FSU College of Engineering, in 2016. She has been a Faculty Advisor of Women in STEM, since 2016, and our NSBE Pre-Collegiate, since 2020, chapters. She has also been a Former Faculty Advisor of SWE, since 2015. In addition, she serves on the Board of a Local Community Organization, STEM4Girls, Incorporated.

SIMON Y. FOO received the B.S., M.S., and Ph.D. degrees in electrical and computer engineering from the University of South Carolina, in 1983, 1985, and 1988, respectively. He joined Florida A&M University and Florida State University, in 1988, where he is currently a tenured Professor with the Department of Electrical and Computer Engineering. He was formerly the ECE Department Chair, from 2010 to 2018. His research contributions have been in the areas of machine learning and high-efficiency multi-junction III–V compound and polymer solar cells. He has authored or coauthored over 100 refereed technical articles, was awarded two patents on high-efficiency multi-junction and quantum dot solar cells, and contributed to two book chapters. He has also graduated more than 50 M.S. and Ph.D. students. He is also the Principal Investigator of more than 20 funded research projects with the Office of Naval Research, National Security Agency, National Science Foundation, Air Force, Boeing Aircraft Company, and Florida Department of Transportation. He was a Technical Advisor to Airbus in the field of fire detection in aircraft cargo bays and as an Expert Witness in VLSI design. He has been a member of Eta Kappa Nu Electrical Engineering Honor Society, since 1982. He has won several awards, including the ''Engineering Research Awards'' from the FAMU-FSU College of Engineering (twice), the Engineering Honor Society Tau Beta Pi ''Teacher of the Year'' Award, and ''Best Paper'' awards in international conferences.

FANGYUAN SUN (Student Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the School of Electrical Automation and Information Engineering, Tianjin University, China, in 2019 and 2022, respectively. He is currently pursuing the Ph.D. degree with the College of Electrical Engineering, Zhejiang University. His research interest includes the application of artificial intelligence in power systems.

AYODEJI S. OGUNDANA (Graduate Student Member, IEEE) received the B.S. degree in mechanical engineering from The Federal University of Technology Akure (FUTA), Nigeria, in 2012, and the M.S. degree in electrical engineering from Florida State University, in 2019, where he is currently pursuing the Ph.D. degree. His research interests include machine learning and control systems for electric-vehicle battery management systems.