RESEARCH ARTICLE

Deep Neural Network-Based Surrogate Model for Optimal Component Sizing of Power Converters Using Deep Reinforcement Learning

VAN-HAI BUI[1], (Member, IEEE), FANGYUAN CHANG[1], (Graduate Student Member, IEEE), WENCONG SU[1], (Senior Member, IEEE), MENGQI WANG[1], (Senior Member, IEEE), YI LU MURPHEY[1], (Life Fellow, IEEE), FELIPE LENO DA SILVA[2], (Member, IEEE), CAN HUANG[2], (Senior Member, IEEE), LINGXIAO XUE[3], (Senior Member, IEEE), AND RUBEN GLATT[2], (Member, IEEE)

[1]Department of Electrical and Computer Engineering, College of Engineering and Computer Science, University of Michigan-Dearborn, Dearborn, MI 48128, USA
[2]Lawrence Livermore National Laboratory (LLNL), Livermore, CA 94550, USA
[3]Oak Ridge National Laboratory (ORNL), Oak Ridge, TN 37830, USA

Corresponding author: Wencong Su (wencong@umich.edu)

ABSTRACT The optimal design of power converters often requires a huge number of simulations and numeric analyses to determine the optimal parameters. This process is time-consuming and results in a high computational cost. Therefore, this paper proposes a deep reinforcement learning (DRL)-based optimization algorithm to optimize the design parameters for power converters using a deep neural network (DNN)-based surrogate model. The surrogate model of power converters can quickly estimate the power efficiency from input parameters without requiring any simulation. The proposed optimization model includes two major steps. In the first step, the surrogate model is trained offline using a large dataset. In the second step, a soft actor-critic-based optimization model interacts with the surrogate model from step 1 to determine the optimal values of design parameters in power converters. Unlike deep Q learning-based methods, the proposed method is able to handle large state and action spaces. In addition, using entropy-regularized reinforcement learning, our proposed method can accelerate and stabilize the learning process and also prevent trapping in local optima. Finally, to show the effectiveness of the proposed method, the performance of different optimization algorithms is compared, considering over ten power converter topologies.

INDEX TERMS Component sizing, deep reinforcement learning, deep neural networks, optimal design parameters, optimization, power converters, surrogate model.

NOMENCLATURE
ABBREVIATIONS
DNN	Deep neural network
DRL	Deep reinforcement learning
DDPG	Deep deterministic policy gradient
SAC	Soft actor-critic

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

I. INTRODUCTION

With the rapid growth of clean energy sources (wind turbines and PV panels) and their integration with the power system, hybrid AC-DC power networks are becoming popular in modern power systems. Due to the integration of AC/DC sources and AC/DC loads, a DC subgrid has recently been developed and tied to the AC bus via one or multiple bidirectional DC/AC interlinking converters [1]. In a subgrid, primary energy resources and local load are commonly

connected to the electrical network via DC-DC/AC-DC converters [2].

Power converters consist of interconnected individual circuit components (e.g., resistors, capacitors, inductors, diodes, and switching devices), making their design quite complex and prone to inefficiencies. To increase the energy conversion efficiency and stability in modern power systems, power converters are often optimally designed to achieve different operation goals [3]–[6]. For instance, the authors in [4] have proposed a novel soft-switching integrated boost DC-DC converter for PV power systems to reduce loss and improve power conversion efficiency. The authors in [5] have developed an optimization-based design method for high-efficiency power converters that considers the power factor and limitations of current ripples. The authors in [6] have developed a multi-objective design for buck converters to improve the overall performance of the output LC filter.

To achieve the design goals, many optimization algorithms have also been proposed and widely used for specific problem, including single- or multiple-objective problems [7], [8]. The simulated annealing algorithm has been utilized for parameter identification of an energetic hysteresis model [9]. A population-based search algorithm, the bee colony optimization algorithm, has been used for a Sheppard-Taylor power factor correction converter [10]. A genetic algorithm and particle swarm optimization algorithm are proposed in [11] for the total power loss optimization of a resonant converter. A novel optimal design for medium-frequency transformers using a multi-objective genetic algorithm has been presented in [12] to determine the best transformer for a given power converter topology.

However, these optimization algorithms [8]–[12] have often been developed based on a model-based approach. Therefore, a complex model might be required, and it may be impossible to develop detailed models for complicated power converter topologies. Therefore, several recent studies have applied self-learning models, e.g., reinforcement learning (RL), for decision-making problems of power converter design [13]–[18]. In RL, there is no requirement for any model of the dynamic environment, and the learning agent evaluates its selected actions based on penalties or rewards obtained by interacting with the dynamic environment [19], [20]. A Markov decision process (MDP) model and a deep Q network-based optimization model have been developed in [15] for the stabilization issue of power converters. A Q-learning algorithm has been applied in [16] to train an agent offline for a DC-DC converter, and then the trained agent provides control decisions online during real-time operation to reduce the power losses. A deep deterministic policy gradient (DDPG) algorithm has been applied in [17] to minimize power losses in dual active bridge (DAB) converters. A DDPG-based optimization model has been proposed in [18] to adjust the active disturbance rejection controller for the voltage regulation of DC-DC buck converters.

Most of these studies have focused on solving optimal parameter design problems using Q-learning, deep Q-learning [15], [16], or DDPG [17], [18]. In these methods, it is often difficult to tune hyperparameters to achieve global optimization due to brittle convergence properties and high sample complexity [21], [22]. Using Q learning and deep Q network-based models also present challenges to handling a large or continuous action space [23], [24]. In addition, in the optimal parameter design for power converters, thousands of simulations are required for the optimization process. This process often takes many hours to determine the optimal solution with high computation cost. Therefore, an approximation model to replace dynamic simulation modeling is necessary to quickly map the input information onto the output measurement.

To overcome the above-mentioned problems, a soft actor-critic (SAC)-based optimization algorithm is proposed in this study to optimize the design parameters of power converters. First, a DNN-based surrogate model is developed and is then trained by an offline process. Then, the trained agent can interact with the surrogate model to quickly estimate the power efficiency, without requiring any simulation. This can significantly reduce the learning time for optimal parameter design of power converters. Furthermore, the proposed SAC-based optimization model inherits several desirable traits, such as the ability to handle continuous state and action spaces, accelerate and stabilize the learning process, and prevent trapping in local optima by using entropy-regularized RL. The performance of the proposed method is also evaluated using over ten DC-DC power converter topologies. Finally, the performance of the proposed method is compared with different parameter design algorithms for power converters. The major contributions of the paper are as follows.

- An SAC-based optimization strategy is proposed to optimize the parameter design for power converters using a deep neural network-based surrogate model. The use of entropy-regularized reinforcement learning results in an accelerated and stabilized learning process.
- Deep neural network-based surrogate models are developed to replace dynamic simulation models. This greatly reduces the learning time for the DRL agent to find the optimal solution.
- Implementation of the state-of-the-art RL algorithm can achieve a better cumulative reward and can determine better solutions for the parameter design problems. The proposed method also shows good performance with over ten test topologies.

The rest of the paper is organized as follows. Section II discusses the development of a DNN-based surrogate model for power converters. Section III presents the proposed DRL-based optimization algorithm for component sizing of power converters. Section IV shows the effectiveness of the proposed method with various
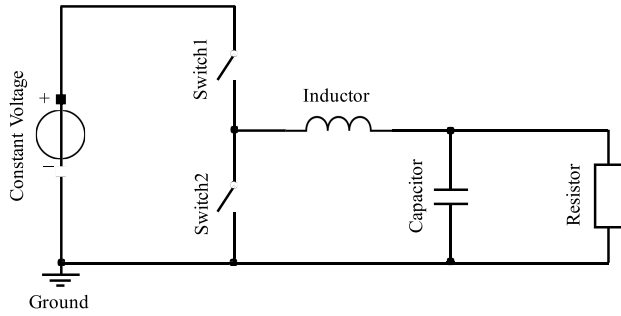
(a). Buck topology.
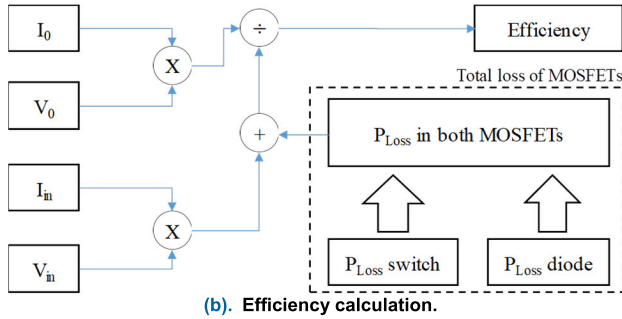
(b). Efficiency calculation.

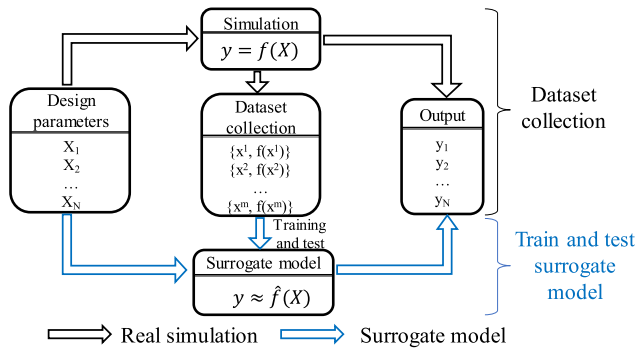**FIGURE 1.** A test case with buck converter.

FIGURE 2. Typical surrogate model.

power converter topologies. Finally, conclusions are drawn in Section V.

II. SURROGATE MODEL FOR POWER CONVERTERS
A. PHYSICAL MODEL AND SURROGATE MODEL
1) PHYSICAL/DYNAMIC MODEL
Figure 1(a) shows a popular power converter topology used in power systems (i.e., buck converter). This topology consists of a DC voltage source, two switches, a capacitor, and a resistor. Figure 1(b) describes a block diagram to calculate the power efficiency using input/output voltage, input/output current, and power losses. In the design process of power converters, the values of input parameters were varied within a predefined boundary and power efficiency was measured by carrying out Simulink simulations. However, this process might require hundreds of thousands of simulations for a single user's desired outputs. Therefore, we collect simulation data and use it to train a surrogate model to reduce the design cycle of power converters, as shown in Figure 2.
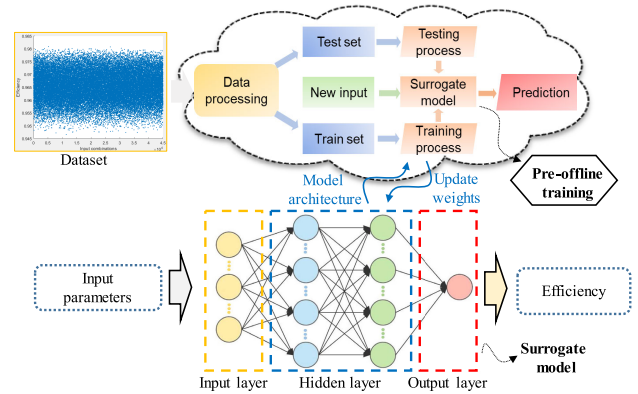

FIGURE 3. DNN-based surrogate model.

2) SURROGATE MODEL
The surrogate model is an approach used when the outcomes of a model cannot be directly measured easily or require a long time to complete a simulation [25], [26]. The surrogate model is developed in those cases to mimic the behaviors of the physical or simulation models as closely as possible [26]. This will greatly reduce the time spent in optimizing the design and analyzing the sensitivity of a model that requires thousands of simulations to carry out.

Figure 2 shows the process of developing a surrogate model. This process is divided into two main stages, (i) dataset collection and (ii) training and testing of the surrogate model. The dataset is generated by performing simulations with different setting values of the input parameters $\{x_1, x_2, \ldots, x_N\}$. The output signal is measured at the end of the simulation $\{y_1, y_2, \ldots, y_N\}$. This process is executed iteratively to obtain a large dataset for the training and testing processes of the surrogate model. After the training and testing processes, the surrogate model is used to quickly map the input parameters onto the output with high accuracy.

B. DNN-BASED SURROGATE MODELS FOR POWER CONVERTERS
In this study, a DNN-based surrogate model is used to replace the simulation model of power converters. Firstly, a large number of simulations are performed using MAT-LAB/Simulink with different combinations of input parameters, such as resistance, inductance, switching frequency, and so on. This dataset is then split into training and test sets for training the surrogate model. After the training process, the surrogate model can provide an estimation of the power efficiency that is close to the Simulink simulation results using the same input information. The training and testing process of the DNN-based surrogate models of the power converter is summarized in Figure 3.

Table 1 shows ten common parameters for a typical power converter (e.g., buck converter). This table provides setting-values boundaries for different components like inductors, capacitors, resistors, and switching components. The number of converter parameters determines the number of nodes in the input layer or the size of the input layer. For instance,

TABLE 1. Input information of surrogate model for buck converter.

No.	Parameters	Operation bounds
1	Switching frequency f_s (Hz)	[2e4, 1e6]
2	Inductance L (H)	[5e-7, 1e-4]
3	Capacitance C (F)	[1e-5, 5e-4]
4	MOSFET resistance (Ω)	[1e-4, 1e-2]
5	Diode voltage V_d (V)	[0.1, 1]
6	Rising time t_r (s)	[1e-10, 30e-9]
7	Falling time t_f (s)	[1e-10, 20e-9]
8	Input voltage V_{in} (V)	[28, 48]
9	Output voltage V_{out} (V)	[5, 20]
10	Output power P_{out} (W)	[10, 20]

TABLE 2. Setup of DNN-based surrogatge model.

Parameters	Setting values
Input size	10
Output size	1
Learning rate	0.001
Hidden layer 1	256
Hidden layer 2	256
Activation function	Relu
Optimizer	Adam
Batch size	64
Loss	MSE

10 input parameters are considered for a buck converter, so the input layer size is 10 nodes. The different values of input parameters are fed into the input layers of the DNN-based surrogate model. The output of the surrogate model represents the power efficiency of the converter and the output size is always 1. To determine the optimal parameters for DNN, a sensitivity analysis was performed with many different network configurations. Selected configuration not only avoids underfitting and overfitting issues but also minimizes the training/testing loss. Detailed parameters for the DNN-based surrogate model are tabulated in Table 2.

III. DRL-BASED COMPONENT SIZING OF POWER CONVERTERS

In this section, a detailed DRL-based optimization model is presented for optimal component sizing of the power converter. In power converter design, there are two major steps including (i) designing the converter topology and (ii) optimal component sizing of a given topology. This study mainly focuses on step 2 to solve the component sizing problem. Since a power converter is composed of various components, different sizes of components might significantly affect the performance of the power converter (e.g., power efficiency). Therefore, optimal component sizing for power converters is an important task to determine the suitable size of all components in a power converter for a given design objective, such as reducing design costs or increasing power efficiency.

A. COMPONENT SIZING AS A MARKOV DECISION PROCESS

Here, we formulate the component sizing problem of power converter as an MDP which consists of four essential

elements: state, action, state transition, and reward, as follows [19], [20].

- State (s_t): A state represents the information about setting values of input parameters of power converter, as shown in (1). The value of state parameters needs to be normalized for the convergence and stability of the training process, as given in (2).
- Action (a_t): The agent observes the system state s_t at time t and selects an action, which represents the change in input design parameters, as shown in (3).
- State transition: The state's transition from time t to $t + 1$ is shown as $s_{t+1} = f(s_t, a_t)$ and the state transition mainly depends on the action selection.
- Reward (r_t): A reward function represents the improvement in power efficiency at the current state compared with the initial state, as shown in (4), and the learning agent attempts to maximize the cumulative reward during the training process. The reward at each state is quickly determined using the surrogate model.

$$s_t = \left[P^{Set}_{m,t}, \ldots, P^{Set}_{M,t} \right] \quad \forall t \in T \tag{1}$$

$$s_t[k] = \frac{s_t[k] - s_t^{min}[k]}{s_t^{max}[k] - s_t^{min}[k]} \quad \forall k \in K, \ t \in T \tag{2}$$

$$a_t = \{ \Delta P^{Set}_{m,t}, \ldots, \Delta P^{Set}_{M,t} \} \tag{3}$$

$$r_t = SM(s_t + a_t) - SM(s_{inital}) \quad \forall t \in T \tag{4}$$

A SAC-based optimization model is implemented to determine the optimal components' size of power converter.

B. SAC-BASED OPTIMIZATION MODEL
In this study, soft actor-critic (SAC) is used to optimally determine the component sizing of power converters due to its fast learning and stable training process. The main idea behind SAC is to use a modified RL objective by adding entropy regularization, as given in (5) [22], [27].

$$J(\theta) = \sum_{t \in T} \mathbb{E}_{(s_t, a_t) \sim \rho_{\pi_\theta}} [r(s_t, a_t) + \alpha \mathcal{H}(\pi_\theta(.|s_t))] \tag{5}$$

where:

$\mathcal{H}$ is the entropy measure

α is a weight factor (or temperature parameter) representing the importance of the entropy term

The entropy term indicates how unpredictable a random variable is. This means that high entropy leads to the value of the variable being more unpredictable. Therefore, high entropy is important to explicitly encourage exploration, and the policy assigns the same probability to actions having the same Q-values. This ensures that the learning agent does not collapse into repeatedly selecting the same action during the learning process.

In SAC, the learning agent learns a policy network $\{\pi_\theta\}$ and two Q-networks $\{Q_{\phi_1}, Q_{\phi_2}\}$. The loss function of the Q-networks is calculated using (6), while the target is computed in (7). The Q-function is updated by one step of gradient

descent using (8).

$$L(\phi_i, \mathcal{D}) = \underset{(s,a,r,s',d) \sim \mathcal{D}}{E} \left[\left(Q_{\phi_i}(s, a) - y(r, s', d) \right)^2 \right] \quad (6)$$

$$y(r, s', d) = r + \gamma(1-d)\Big(MinQ_{\phi_{tar,i}}(s', \tilde{a}')$$
$$- \alpha log\pi_\theta(\tilde{a}'|s')\Big) \quad i = 1, 2 \quad (7)$$

$$G_\phi = \nabla_{\phi_i} \frac{1}{|B|} \sum_{sample\ in\ B} \left(Q_{\phi_i}(s, a) - y(r, s', d) \right)^2$$
$$i = 1, 2 \quad (8)$$

The learning agent tries to maximize the sum of the expected future returns and expected future entropy. This helps the learning agent to accelerate the learning process and avoid trapping in local optima. The policy network is updated by one step of gradient ascent using (9). Finally, the target network is updated by soft copy, as given in (10). This helps the target network to change slowly and ensures a stable learning process.

$$G_\theta = \nabla_\theta \frac{1}{|B|} \sum_{sample\ in\ B} \Big(MinQ_{\phi_i}(s, \tilde{a}_\theta(s))$$
$$- \alpha log\pi_\theta(\tilde{a}_\theta(s)|s)\Big) \quad i = 1, 2 \quad (9)$$

$$\phi_{tar,i} \leftarrow \rho\phi_{tar,i} + (1-\rho)\phi_i \quad i = 1, 2 \quad (10)$$

The optimization algorithm and learning process are presented in detail in the following section.

C. OPTIMIZATION ALGORITHM AND LEARNING PROCESSES

The optimization process for maximizing power efficiency is shown in Figure 4 with two major learning models, including training a surrogate model and the optimal parameter design of power converters.

Firstly, a DNN-based surrogate model is trained to mimic the behavior of a given power converter with a large dataset. This dataset is collected by carrying out a Simulink simulation, where input combinations are randomly selected and measure a simulation outcome (i.e. power efficiency). A DNN with optimal weights is used to predict the power efficiency for the optimal components sizing of the power converter.

A DRL agent is developed to optimize the component size of a power converter using the soft actor-critic algorithm. Firstly, five different deep neural networks are initialized for the learning process, including a policy network, two Q networks, and two target networks. A memory of size $\mathcal{D}$ is also initialized for the experience replay process. The learning agent starts at the initial state with initial setting values for all components and chooses actions using the current policy. The agent carries out these actions, receives an immediate reward, and observes a new state. The immediate reward is determined using the DNN-based surrogate model, instead of carrying out a long simulation (up to several minutes). Optimizing parameter design of power converters typically requires a large number of simulations. Therefore,

the use of the surrogate model significantly reduces the learning time for the DRL agent. After finishing a transition from s to s', the information of $\{s, a, r, s'\}$ is stored in the memory $\mathcal{D}$. During a training step, a mini-batch is randomly drawn from $\mathcal{D}$ and the target y, and losses are then calculated. The optimization step is carried out to minimize the loss function. The weights of all deep neural networks are updated using gradient descent (i.e., Q-functions) and gradient ascent (i.e., policy). The target network is slowly updated using a soft copy from Q-functions; this helps to stabilize the learning process. The learning process is summarized in Algorithm 1 and is repeated for multiple episodes.

Algorithm 1: SAC-Based Optimal Component Sizing of Power Converters

---

Initialize policy θ, Q-function ϕ_1, ϕ_2, replay buffer $\mathcal{D}$
Set target parameters: $\phi_{tar,1} \leftarrow \phi_1$, $\phi_{tar,2} \leftarrow \phi_2$
for $e = 1, E$ **do**: # Learning episodes
    for $k=1, K$ **do**: # Learning step
        Observe state of power converter:
        s is initial values of all components
        Select action $a \sim \pi_\theta(.|s)$ and carry out a to adjust the setting values
        Observe s', reward r, done signal d, and store (s, a, r, s', d) in $\mathcal{D}$
    end
    for $g=1, G$ **do**: # Gradient step
        Randomly sample a batch of transitions B from $\mathcal{D}$
        Compute target Q network:
        $y(r, s', d) =$
        $r + \gamma.(1-d)\Big(Min\ Q_{\phi_{tar,i}}(s', \tilde{a}') - \alpha.log\pi_\theta(\tilde{a}'|s')\Big)$
        $i = 1, 2$
        Update Q network using:
        $\nabla_{\phi_i} \frac{1}{|B|} \sum_{sample\ in\ B} \left(Q_{\phi_i}(s, a) - y(r, s', d) \right)^2$
        $i = 1, 2$
        Update policy network using:
        $\nabla_\theta \frac{1}{|B|} \sum_{sample\ in\ B} \left(Min\ Q_{\phi_i}(s, \tilde{\alpha}_\theta(s)) - \alpha.log\pi_\theta(\tilde{a}_\theta(s)|s) \right)$ $i = 1, 2$
        Update target networks:
        $\phi_{tar_i} \leftarrow \rho.\phi_{tar,i} + (1-\rho).\phi_i$ $i = 1, 2$
    end
end

After training with a large number of episodes, the learning agent can use policy parameters to determine the optimal setting values for components of power converters with any initial input parameters.

IV. SIMULATION RESULTS
A. SURROGATE MODEL: TRAINING AND TESTING PROCESSES

In this section, the detailed training and evaluation process of a DNN-based surrogate model is presented for a specific case (i.e., buck topology). The same process is performed for ten other common topologies, including boost, buck-boost, LLC, DAB, flyback, forward, push-pull, half-bridge, full-bridge, and switched-capacitor converters.

1) BUCK CONVERTER
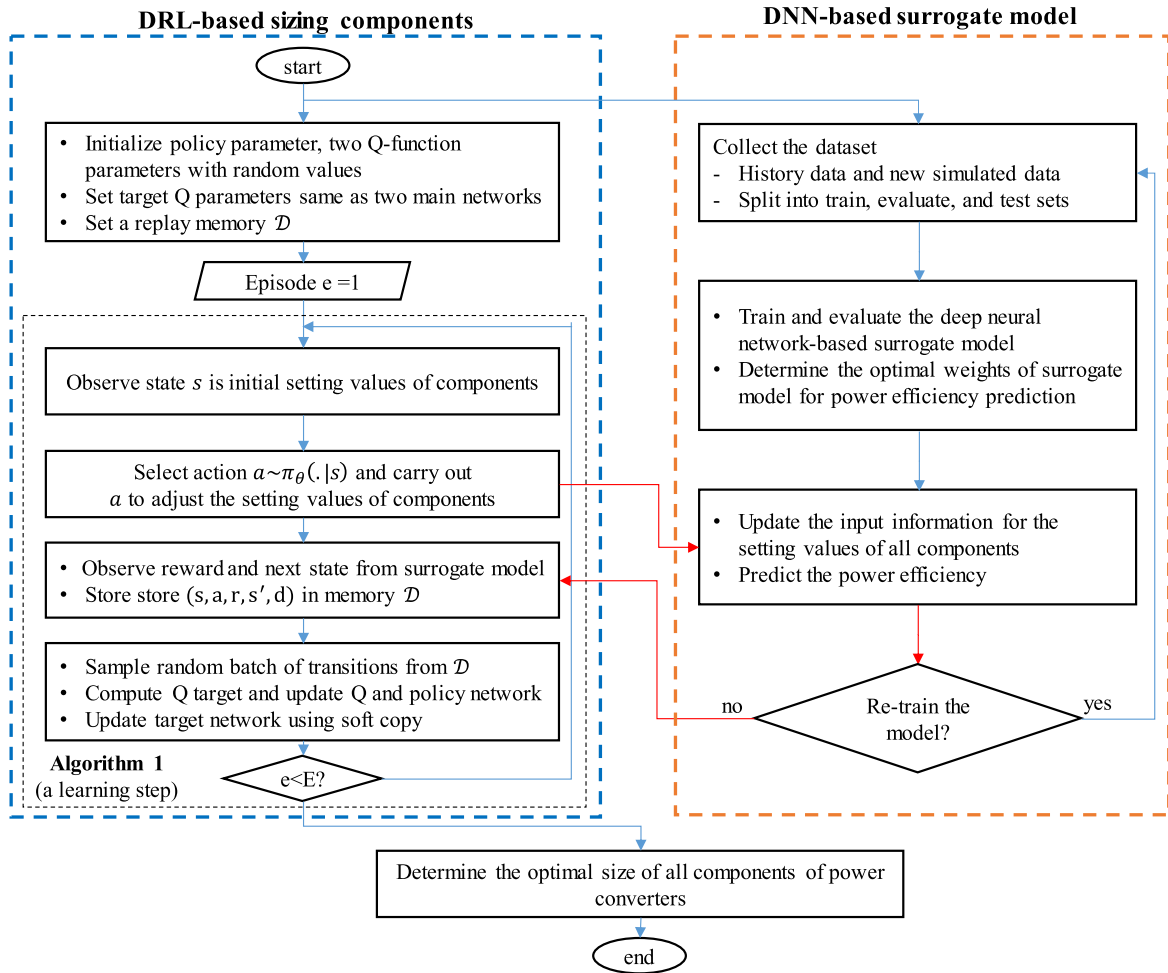To show the performance of the proposed surrogate model, a simple buck converter is used in this section. As mentioned

DRL-based sizing components

DNN-based surrogate model

start

- Initialize policy parameter, two Q-function parameters with random values
- Set target Q parameters same as two main networks
- Set a replay memory $\mathcal{D}$

Episode e =1

Observe state s is initial setting values of components

Select action $a \sim \pi_\theta(.|s)$ and carry out a to adjust the setting values of components

- Observe reward and next state from surrogate model
- Store store (s, a, r, s', d) in memory $\mathcal{D}$

- Sample random batch of transitions from $\mathcal{D}$
- Compute Q target and update Q and policy network
- Update target network using soft copy

Algorithm 1
(a learning step)

e<E?

Collect the dataset
- History data and new simulated data
- Split into train, evaluate, and test sets

- Train and evaluate the deep neural network-based surrogate model
- Determine the optimal weights of surrogate model for power efficiency prediction

- Update the input information for the setting values of all components
- Predict the power efficiency

no · · · Re-train the model? · · · yes

Determine the optimal size of all components of power converters

end

FIGURE 4. Overall flowchart for optimal component sizing with surrogate model.

earlier, a large dataset is required to train and test the surrogate model. In this study, the dataset is generated by carrying out Simulink simulations with different setting values of input information and measuring power efficiency in each case. We consider ten different parameters for the buck converter, as given in Table 1. Therefore, the input and output size of the DNN-based surrogate model are as follows.

(i) Input layer: 10 nodes representing 10 input parameters
(ii) Output layer: 1 node representing the power efficiency

Other parameters of DNN is optimally selected as given in Table 2. Firstly, a sensitivity analysis is performed to determine a suitable size for the dataset. The DNN-based surrogate model is trained with different sizes of datasets, and the average error on the test set is summarized in Figure 5. It can be observed that with a larger dataset, the performance of the surrogate model is improved. However, generating a large dataset is time-consuming, so a trade-off between the size of the dataset and the prediction error is considered. A suitable size for the dataset is about 30,000 data points, and the error of the proposed model is also within an acceptable range. Furthermore, with a larger dataset (i.e. over 30,000), the performance of the surrogate model does
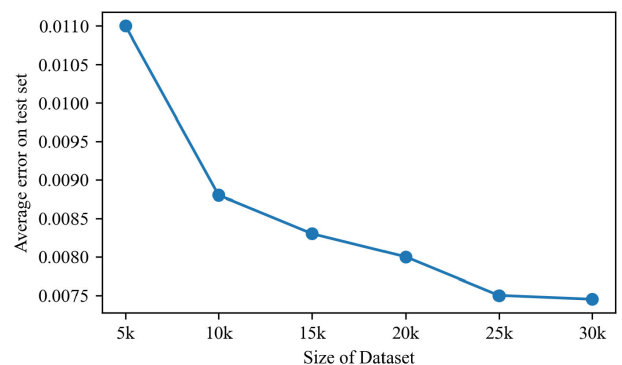

FIGURE 5. Performance of surrogate model with different sizes of dataset.

not show significant improvement. Thus, a dataset of 30,000 is used for all DNN-based surrogate models in this study. The detailed dataset for a buck converter is shown in Figure 6. With random combinations of 10 input parameters, the power efficiency ranges from [0.35, 1.00].

The dataset is split into the train, validate, and test sets composed of 70%, 20%, and 10% of the dataset respectively. Figure 7 shows the training process for the DNN-based
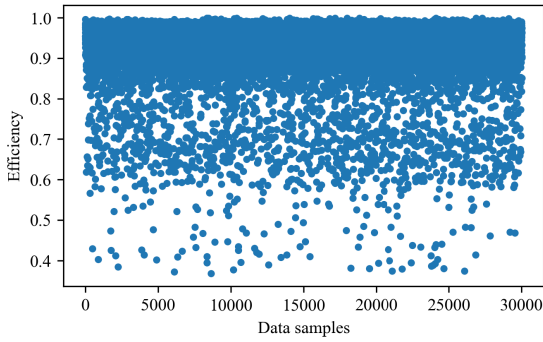
FIGURE 6. Dataset of buck converter with 30,000 samples.

surrogate model. It can be observed that the training and validation losses have converged after 500 training epochs, as shown in Figure 7(a). The well-trained parameters of the DNN are saved for power efficiency prediction. To represent the accuracy of the proposed surrogate model, 100 data points are randomly drawn from the test set and fed to the model. The actual and predicted power efficiencies are shown in Figure 7(b). The detailed error for each data sample is shown in Figure 7(c). It can be observed that the proposed DNN-based surrogate model can estimate the output with acceptable errors. The average error for 100 random test samples is less than 0.005, where the error of each sample and average error of a test set are determined by (11), (12), respectively.

$$e_k = \left| \eta_{act,k} - \eta_{pred,k} \right| \quad (11)$$

$$e_{avg} = \frac{1}{N} \sum_{k=1}^{N} \left| \eta_{act,k} - \eta_{pred,k} \right| \quad (12)$$

where:

|	
$\eta_{act,k}$:	actual power efficiency using Simulink simulation
$\eta_{pred,k}$:	predicted power efficiency using surrogate model
e_{avg}:	average prediction error over N test samples

Finally, the performance of the proposed surrogate model is summarized in Figure 8. Figure 8(a) shows the prediction error with the entire available dataset, while Figure 8(b) shows the prediction error with only new data (i.e., test sets). It can be observed that most test data have a rather small prediction error in $[-0.03, 0.03]$.
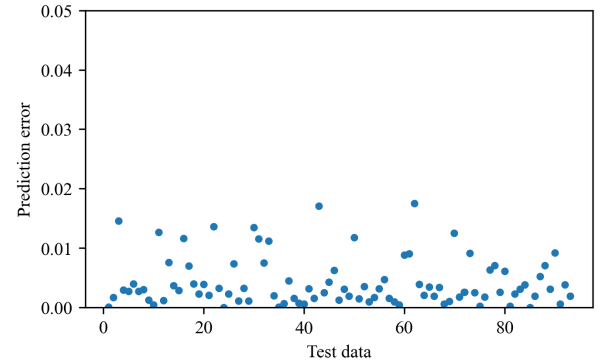
2) OTHER POWER CONVERTERS
As mentioned earlier, the proposed method is tested with 11 well-known converter topologies. Table 3 summarizes the prediction error for DNN-based surrogate models for buck, boost, buck-boost, LLC, DAB, flyback, forward, push-pull, half-bridge, full-bridge, and switched-capacitor converters. It can be observed that all proposed surrogate models are able to provide good performance. Most predictions on the test data have small errors, between 0 to 0.03.
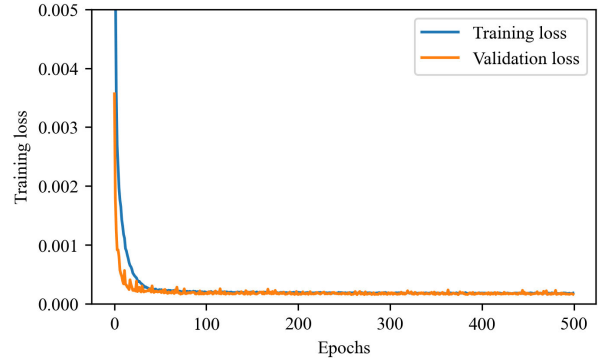
(a). Training and validation losses.
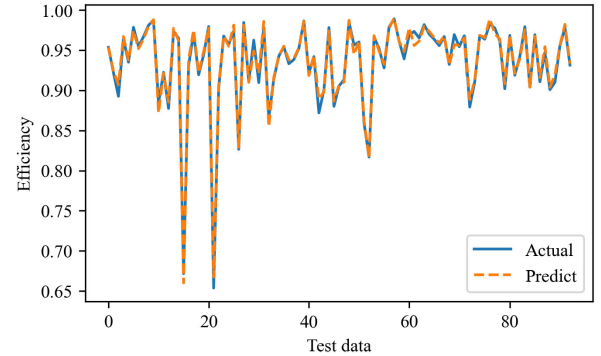
(b). Test random 100 samples in test set.

**(c).** Detailed prediction errors (100 test samples).

FIGURE 7. Training and testing processes of the surrogate model.

### B. DRL-BASED COMPONENT SIZING OF POWER CONVERTERS
In the previous section, DNN-based surrogate models were developed and tested with various power converter topologies to predict power efficiency with high accuracy and also to reduce the learning time for an optimal components sizing process. This section determines the optimal size of components in power converters that use different deep reinforcement algorithms for maximizing the power efficiency.

1) THREE MAJOR INPUT PARAMETERS
Firstly, the optimal number of parameters needs to be specified. In the first test case, only 3 parameters are selected for the optimization process, including switching frequency, inductance, and capacitance. To show the effectiveness of the proposed method, the performance of both DDPG-based
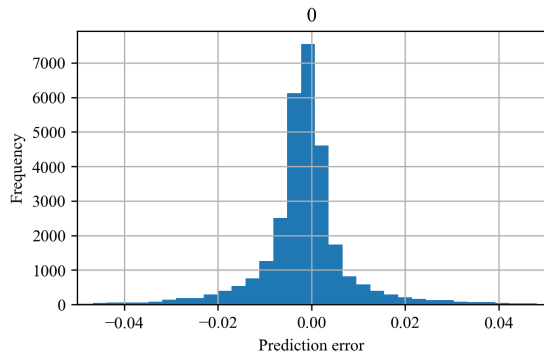
(a). Test with whole dataset (30,000 samples).
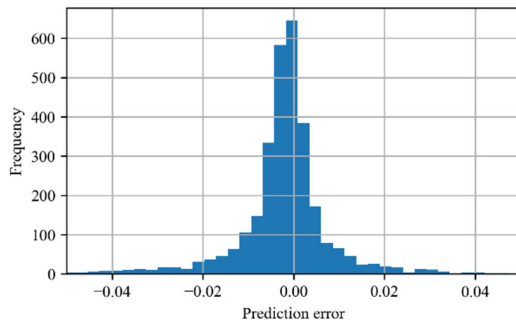
(b). Testing with new data (3,000 samples).

**FIGURE 8.** Testing performance of the surrogate model.

TABLE 3. Test performance of proposed surrogate models (11 topologies).

Parameters	Buck	Boost	Buck-Boost	LLC	DAB	Fly-back
$e<=0.03$	96.60%	96.50%	99.90%	100%	100%	99.30%
$0.03<e<=0.05$	3.20%	2.70%	0.10%	0	0	0.60%
$e>0.05$	0.20%	0.80%	0	0	0	0.10%
Avg. error	0.0075	0.0063	0.0029	0.0027	0.0012	0.0018

| Parameters	Forward	Push-pull	Half-bridge	Full-bridge	Swt. Capacitor
$e<=0.03$	99.30%	100%	100%	100%	98.90%
$0.03<e<=0.05$	0.60%	0	0	0	0.7
$e>0.05$	0.10%	0	0	0	0.4
Avg. error	0.0027	0.0001	0.0007	0.0004	0.0031

and SAC-based optimization models is presented in this study. The training process with both algorithms is shown in Figure 9 for a buck converter. Figure 9(a) shows the total episode reward and Figure 9(b) shows the average rewards with 500 training episodes. It can be observed that both algorithms converge to a close value of total episode rewards after 500 training episodes.

Since both algorithms converge to a similar total reward, the optimal solutions using 2 algorithms are also similar. In addition, the brute force method is also used to search the optimal parameters by performing Simulink simulation with a large number of input combinations. Optimal parameters determined using different methods are presented as follows:

(i)   Brute force: Optimail parameters [F, L, C] are [128889, 0.0001, 0.0005] and optimal efficiency is 0.975592.
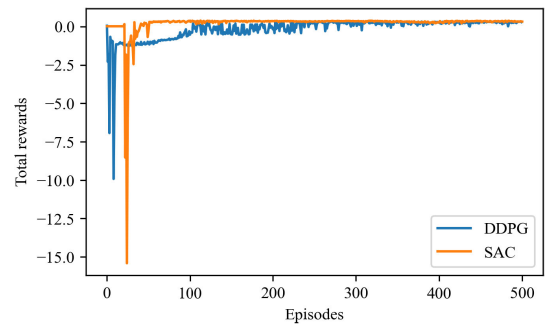
(a). Total reward during 500 training episodes.
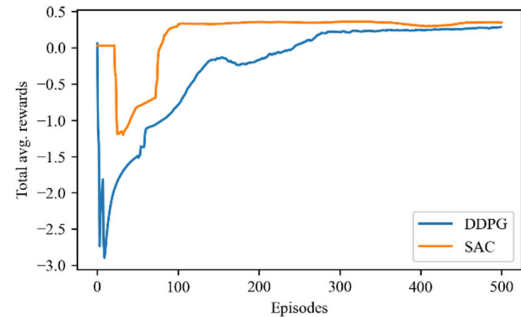
(b). Average reward during 500 training episodes.

FIGURE 9. Training process for three parameters (buck).

(ii)   DDPG: Optimail parameters [F, L, C] are [158964.0, 0.0001, 0.0005] and optimal efficiency is 0.9774.

(iii)   SAC: Optimal parameters [F, L, C] are [148964.0, 0.0001, 0.000185] and optimal power efficiency is 0.9775.

Other parameters are kept constant at default value as follows.

- MOSFET resistance (R_{ds}): 5e-3 Ω
- Diode voltage (V_d): 0.8 V
- Rising time (t_r): 4e-9 s
- Falling time (t_f): 6e-9 s
- Input voltage (V_{in}): 48 V
- Output voltage (V_{out}): 5 V
- Output power (P_{out}): 20 W

Based on the comparison of simulation results of 3 different methods, it can be observed that the proposed method can find the optimal parameters with better power efficiency.

2) INCREASED NUMBER OF INPUT PARAMETERS
In previous case, only 3 parameters are kept constant according to user requirements, including input voltage, output voltage, and output power at 48 V, 5 V, and 20 W, respectively. All remaining parameters are optimally determined using different optimization algorithms in this section. The training process is shown in Figure 10 with 500 episodes. Figure 10(a) shows the total episode rewards, while Figure 10(b) shows the average value of rewards over the learning process. It can be observed that both algorithms converge at high total rewards; however, the SAC-based optimization model can find an optimal solution, which is much better than using the DDPG method.
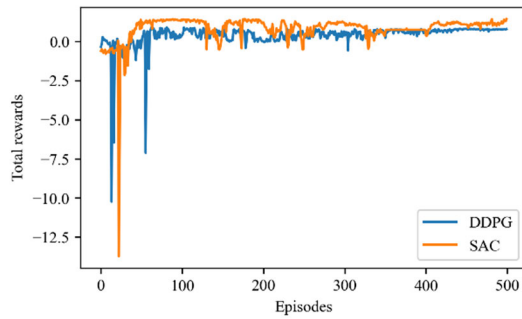
(a). Total reward during 500 training episodes.
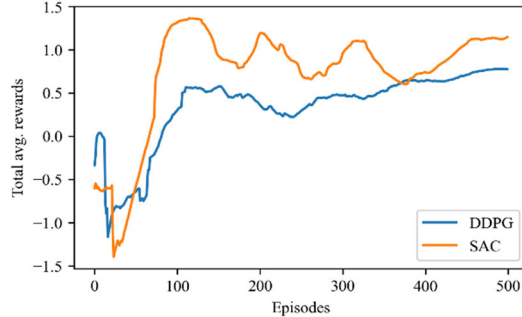
(b). Average reward during 500 training episodes.

FIGURE 10. Training process for seven parameters (buck).

TABLE 4. Optimal parameters using different methods.

Buck	Brute force	Deep Reinforcement Learning	
		SAC	DDPG
Optimal efficiency	0.9959	0.9926	0.9871
Switching frequency f_s (Hz)	174737	277895	174737
Inductance L (H)	2.178e-05	1.0923e-05	6.1374e-06
Capacitance C (F)	0.000465	0.000395	0.000360
MOSFET resistance R_{ds} (Ω)	0.000621	0.00687	0.004789
Diode voltage V_d (V)	0.7	0.1	0.6
Rising time t_r (s)	1e-10	1e-10	1e-10
Falling time t_f (s)	1e-10	1e-10	1e-10
Input voltage V_{in} (V)	48	48	48
Output voltage V_{out} (V)	5	5	5
Output power P_{out} (W)	10	10	10

Table 4 summarizes the results obtained from the three different methods, with detailed information for the setting values of the parameters. The proposed method and brute force simulation can determine better parameters with higher power efficiency than the DDPG-based model. However, the brute force method often requires doing a lot of simulation and this can require a high computational cost.

Similar to the previous section, the proposed method is also tested with different power converter topologies. The optimal solutions for over ten different topologies are presented in

TABLE 5. Optimal solutions for different 11 topologies.

Efficiency	Buck	Boost	Buckboost	LLC	Fly-back
Brute force	0.996	0.999	0.996	0.914	0.925
DRL(SAC)	0.993	0.999	0.992	0.899	0.925

Efficiency	DAB	Push-pull	Full-bridge	Half-bridge	Forward	Swt. capacitor
Bruth force	0.985	0.975	0.974	0.981	0.963	0.988
DRL(SAC)	0.981	0.974	0.974	0.981	0.958	0.987

TABLE 6. Computation cost with/without surrogate model.

Buck		With surrogate model	Without surrogate model
Offline	Data generation	24 hours	-
	Train surrogate model	30 minutes	-
Online	Optimal component sizes	30 minutes	18 hours

Table 5. It can be observed that the proposed method can find optimal solutions close to those of the brute force method, where a large number of simulations are carried out with random input combinations.

3) COMPUTATION COST COMPARISON

Table 6 summarizes the computation cost for determining the optimal parameters for power converters. In most studies available in the literature on optimal parameter design, the optimization model typically measures the response from the dynamic environment by performing an actual simulation. Therefore, it often prolongs the learning process considerably. In Table 6, the buck converter is presented as an example, with detailed information for the offline/online learning process. It can be seen that an offline learning process is required to train the surrogate model and it may take about 1 day. However, this is just an offline learning process and the well-trained surrogate model is used to accelerate the online learning process to determine component sizes ($\sim$30 minutes), whereas without a surrogate model, the learning agent must carry out a large number of simulations. This significantly increases the online learning time ($\sim$18 hours). Moreover, this online learning process is required every time if the user information changes (e.g., input voltage, output voltage, output power).

V. CONCLUSION

In this paper, a soft actor-critic (SAC)-based optimization strategy using a deep neural network-based surrogate model is proposed to optimize the design parameters for power converters. Firstly, a DNN-based surrogate model is trained for each power converter with a large dataset. After the offline training process, the online training time is significantly reduced to only about 30 minutes with any change in user settings, whereas it may take up to 18 hours to determine the optimal solution with each change in user settings without the surrogate model. In addition, the comparison of different optimization algorithms shows that the SAC-based

models can help the learning agent converge quickly to the optimal solution. The results obtained using the SAC-based optimization model are acceptable and are similar to the brute force method, i.e., carrying out a large number of simulations with random input combinations.

ACKNOWLEDGMENT

REFERENCES

[1] E. Unamuno and J. A. Barrena, "Hybrid AC/DC microgrids—Part I: Review and classification of topologies," *Renew. Sustain. Energy Rev.*, vol. 52, pp. 1251–1259, Dec. 2015.

[2] S. K. Sahoo, A. K. Sinha, and N. K. Kishore, "Control techniques in AC, DC, and hybrid AC–DC microgrid: A review," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 6, no. 2, pp. 738–759, Jun. 2018.

[3] J. Saha, N. B. Y. Gorla, A. Subramaniam, and S. K. Panda, "Analysis of modulation and optimal design methodology for half-bridge matrix-based dual-active-bridge (MB-DAB) AC–DC converter," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 10, no. 1, pp. 881–894, Feb. 2022.

[4] K. Sayed, M. G. Gronfula, and H. A. Ziedan, "Novel soft-switching integrated boost DC–DC converter for PV power system," *Energies*, vol. 13, no. 3, p. 749, Feb. 2020.

[5] H. Xu, D. Chen, F. Xue, and X. Li, "Optimal design method of interleaved boost PFC for improving efficiency from switching frequency, boost inductor, and output voltage," *IEEE Trans. Power Electron.*, vol. 34, no. 7, pp. 6088–6107, Jul. 2019.

[6] X. Li, X. Zhang, and F. Lin, "Multi-objective design of output LC filter for buck converter via the coevolving-AMOSA algorithm," *IEEE Access*, vol. 9, pp. 11884–11894, 2021.

[7] S. E. De León-Aldaco, H. Calleja, and J. A. Alquicira, "Metaheuristic optimization methods applied to power converters: A review," *IEEE Trans. Power Electron.*, vol. 30, no. 12, pp. 6791–6803, Dec. 2015.

[8] K. Sundareswaran and V. T. Sreedevi, "Boost converter controller design using queen-bee-assisted GA," *IEEE Trans. Ind. Electron.*, vol. 56, no. 3, pp. 778–783, Mar. 2009.

[9] R. Liu and L. Li, "Simulated annealing algorithm coupled with a deterministic method for parameter extraction of energetic hysteresis model," *IEEE Trans. Magn.*, vol. 54, no. 11, pp. 1–5, Nov. 2018.

[10] A. Karaarslan, "The implementation of bee colony optimization algorithm to Sheppard–Taylor PFC converter," *IEEE Trans. Ind. Electron.*, vol. 60, no. 9, pp. 3711–3719, Sep. 2013.

[11] B. Zhao, X. Zhang, and J. Huang, "AI algorithm-based two-stage optimal design methodology of high-efficiency CLLC resonant converters for the hybrid AC–DC microgrid applications," *IEEE Trans. Ind. Electron.*, vol. 66, no. 12, pp. 9756–9767, Dec. 2019.

[12] A. Garcia-Bediaga, I. Villar, A. Rujas, L. Mir, and A. Rufer, "Multiobjective optimization of medium-frequency transformers for isolated soft-switching converters using a genetic algorithm," *IEEE Trans. Power Electron.*, vol. 32, no. 4, pp. 2995–3006, Apr. 2016.

[13] S. Fan, N. Cao, S. Zhang, J. Li, X. Guo, and X. Zhang, "From specification to topology: Automatic power converter design via reinforcement learning," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, Nov. 2021, pp. 1–9.

[14] D. Alfred, D. Czarkowski, and J. Teng, "Model-free reinforcement-learning-based control methodology for power electronic converters," in *Proc. IEEE Green Technol. Conf. (GreenTech)*, Apr. 2021, pp. 81–88.

[15] C. Cui, N. Yan, B. Huangfu, T. Yang, and C. Zhang, "Voltage regulation of DC–DC buck converters feeding CPLs via deep reinforcement learning," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 3, pp. 1777–1781, Mar. 2022.

[16] Y. Tang, W. Hu, J. Xiao, Z. Chen, Q. Huang, Z. Chen, and F. Blaabjerg, "Reinforcement learning based efficiency optimization scheme for the DAB DC–DC converter with triple-phase-shift modulation," *IEEE Trans. Ind. Electron.*, vol. 68, no. 8, pp. 7350–7361, Aug. 2021.

[17] Y. Tang, W. Hu, B. Zhang, D. Cao, N. Hou, Y. Li, Z. Chen, and F. Blaabjerg, "Deep reinforcement learning-aided efficiency optimized dual active bridge converter for the distributed generation system," *IEEE Trans. Energy Convers.*, vol. 37, no. 2, pp. 1251–1262, Jun. 2022.

[18] M. Gheisarnejad and M. H. Khooban, "IoT-based DC/DC deep learning power converter control: Real-time implementation," *IEEE Trans. Power Electron.*, vol. 35, no. 12, pp. 13621–13630, May 2020.

[19] M. Wiering and M. O. Ma, *Reinforcement Learning State-of-the Art*. Berlin, Germany: Springer-Verlag, 2012.

[20] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, Oct. 2018.

[21] S. Gu, T. Lillicrap, Z. Ghahramani, R. E. Turner, and S. Levine, "Q-Prop: Sample-efficient policy gradient with an off-policy critic," 2016, *arXiv:1611.02247*.

[22] 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*.

[23] V.-H. Bui and W. Su, "Real-time operation of distribution network: A deep reinforcement learning-based reconfiguration approach," *Sustain. Energy Technol. Assessments*, vol. 50, Mar. 2022, Art. no. 101841.

[24] V.-H. Bui, A. Hussain, and W. Su, "A dynamic internal trading price strategy for networked microgrids: A deep reinforcement learning based game-theoretic approach," *IEEE Trans. Smart Grid*, early access, Apr. 20, 2022, doi: 10.1109/TSG.2022.3168856.

[25] A. Bhosekar and M. Ierapetritou, "Advances in surrogate based modeling, feasibility analysis, and optimization: A review," *Comput. Chem. Eng.*, vol. 108, no. 4, pp. 250–267, Jan. 2018.

[26] P. Jiang, Q. Zhou, and X. Shao, *Surrogate Model-Based Engineering Design and Optimization*. Singapore: Springer, 2020.

[27] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2018, pp. 1861–1870.

VAN-HAI BUI (Member, IEEE) received the B.E. degree in electrical engineering from the Hanoi University of Science and Technology, Vietnam, in 2013, and the Ph.D. degree in electrical engineering from Incheon National University, South Korea, in 2020. He is currently a Research Investigator and a Lecturer with the Department of Electrical and Computer Engineering, University of Michigan-Dearborn, USA. His research interests include energy management systems, the applications of reinforcement learning in smart grids, and microgrid operation.

FANGYUAN CHANG (Graduate Student Member, IEEE) received the B.S. degree in electrical engineering from North China Electric Power University, China, in 2016, and the M.S. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2018. She is currently pursuing the Ph.D. degree with the University of Michigan, Dearborn, MI, USA. Her research interests include power systems, control theory, and machine learning.

WENCONG SU (Senior Member, IEEE) received the B.S. degree (Hons.) from Clarkson University, Potsdam, NY, USA, in 2008, the M.S. degree from Virginia Tech, Blacksburg, VA, USA, in 2009, and the Ph.D. degree from North Carolina State University, Raleigh, NC, USA, in 2013. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, University of Michigan-Dearborn, USA. He is a fellow of IET. He is a Registered Professional Engineer (P.E.) with the State of Michigan, USA. His current research interests include power systems, transportation electrification, and cyber-physical systems. He is an Editor of IEEE TRANSACTIONS ON SMART GRID, an Associate Editor of IEEE ACCESS, and an Associate Editor of IEEE DataPort. He is the Guest Editor-in-Chief of IEEE TRANSACTIONS ON SMART GRID—Special Section on Power-Electronics-Enabled Smart Power Distribution Grid. He has authored one book, four book chapters, and over 120 high-quality articles in prestigious international journals and peer-reviewed conference proceedings. He was a recipient of the 2015 IEEE Power and Energy Society (PES) Technical Committee Prize Paper Award and the 2013 IEEE Industrial Electronics Society (IES) Student Best Paper Award. He was a 2021 recipient of the Distinguished Research Award at the University of Michigan-Dearborn. He has been serving as the Campus Director for the Center for Electric Drive Transportation, since 2015, and a NSF REU Site, since 2018, with the University of Michigan-Dearborn. He has also been serving as an Associate Director for the Dearborn Artificial Intelligence Research Center, since 2020.
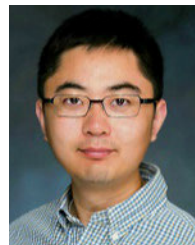
MENGQI WANG (Senior Member, IEEE) received the B.S. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, in 2009, and the Ph.D. degree in electrical engineering from North Carolina State University, Raleigh, NC, USA, in 2014. She is currently an Associate Professor with the Department of Electrical and Computer Engineering, University of Michigan-Dearborn. Her research interests include DC-DC and DC–AC power conversions, high efficiency and high power-density power supplies, renewable energy systems, and wide-bandgap power device applications.

YI LU MURPHEY (Life Fellow, IEEE) received the M.S. degree in computer science from Wayne State University, Detroit, Michigan, in 1983, and the Ph.D. degree with a major in computer engineering and a minor in control engineering from the University of Michigan, Ann Arbor, Michigan, in 1989. She is currently a Professor with the Electrical and Computer Engineering (ECE) Department and the Director of the Intelligent Systems Laboratory and Driving Simulator Laboratory, University of Michigan-Dearborn. She has authored over 290 publications in refereed journals and conference proceedings. Her current research interests include the areas of machine learning, pattern recognition, computer vision and intelligent systems with applications to engineering diagnostics and prognostics, optimal vehicle power management, data analytics, and robotic vision systems. She is a Distinguished Lecturer for both IEEE Vehicular Society and IEEE Computational Intelligence Society.
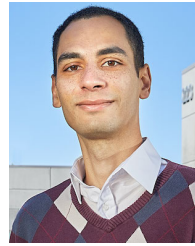
FELIPE LENO DA SILVA (LENO) (Member, IEEE) received the Ph.D. degree in computer engineering from the University of São Paulo, Brazil, in 2019. He has been a Postdoctoral Reinforcement Learning Researcher with the Lawrence Livermore National Laboratory, USA, since 2021, where he applies RL to varied applications. He is currently a Guest Editor of *Neural Computing and Applications*. He has organized several workshop series, including the Adaptive and Learning Agents (ALA) workshop, from 2020 to 2022, and the Latinx in AI workshop at NeurIPS. He has published one book and over 40 scientific papers in peer-reviewed venues. His research interests include several AI areas and applications, such as reinforcement learning, transfer learning, multiagent systems, power electronics, smart grid, transportation, antibody development, and AI-assisted healthcare.

CAN HUANG (Senior Member, IEEE) received the B.S.E.E. degree from Hohai University, Nanjing, China, in 2008, the M.S.E.E. degree from Southeast University, Nanjing, in 2011, and the Ph.D. degree from the University of Tennessee, Knoxville, TN, USA, in 2016, all in electrical engineering. He is currently a Research Staff with the Lawrence Livermore National Laboratory, Livermore, CA, USA. His current research interests include smart sensors, data analytics, and machine learning for energy and power systems, cyber-physical systems, and the Internet of Things.
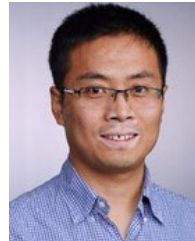
LINGXIAO XUE (Senior Member, IEEE) received the B.S. and M.S. degrees from Zhejiang University, Hangzhou, China, in 2006 and 2008, respectively, and the Ph.D. degree from Virginia Tech, Blacksburg, VA, USA, in 2015, all in electrical and electronics engineering. From 2015 to 2019, he was a Staff Application Engineer at Navitas Semiconductor. He is currently a Research and Development Staff Member and the Subprogram Manager for building electric appliances, devices, and systems (BEADS) with the Oak Ridge National Laboratory. His research interests include advanced power electronics and architecture concepts, including wideband gap semiconductors adoption, high-frequency power conversion, power supply standardization, and wireless power transfer, to cut carbon footprint, improve user experience, and enable grid-interactive. He is an Associate Editor of IEEE OPEN JOURNAL OF POWER ELECTRONICS. He also served in the IEEE Power Electronics Society as the Young Professionals Chair, and the Coastal Los Angeles Section Chapter Chair.

RUBEN GLATT (Member, IEEE) received the Dipl.-Ing. degree in mechatronics from the Karlsruhe Institute of Technology, Karlsruhe, Germany, in 2011, the master's degree in mechanical engineering from the Universidade Estadual Paulista, São Paulo, Brazil, in 2014, and the Ph.D. degree in computer engineering from the University of São Paulo, São Paulo, in 2019. Then, he joined the Lawrence Livermore National Laboratory, California, USA, for a two-year postdoctoral assignment from which he converted to a Full Staff Researcher, in 2021. His research interests include general machine learning, artificial intelligence, and autonomous agents. He is particular interested in reinforcement learning, knowledge transferm, and multi-agent systems.

● ● ●