Machine Learning-Based Control of Electric Vehicle Charging for Practical Distributions Systems with Solar Generation

Ivan Calero, *Member, IEEE,* Claudio A. Cañizares, *Fellow, IEEE,* Mostafa Farrokhabadi, *Senior Member, IEEE* and Kankar Bhattacharya, *Fellow, IEEE,*

***Abstract*—The adoption of Electric Vehicles (EVs) and solar Photovoltaic (PV) generation by households is rapidly and significantly increasing. Utilities are facing the challenge of efficiently managing EV and PV resources to help mitigate the undesirable effects on grid operation. Existing approaches to solve these issues depend on accurate but hard to predict behavior of EVs and PVs, detailed knowledge of customers, and grid infrastructure, all of which complicate the effective deployment of these resources. Motivated by these practical challenges and in collaboration with industry partners working on addressing these issues, this paper proposes a two-level data-driven smart controller for EV charging in distribution systems. The controller is modeled as a Deep Reinforcement Learning (DRL) agent, which coordinates the charging rates of multiple EVs connected to a realistic residential feeder with high penetration of PV generation. The first level coordinates the aggregated EV load at distribution Medium Voltage (MV) level to provide Demand Response (DR) services; at the Low Voltage (LV) level it aims to maximize the EVs' state of charge at departure while avoiding the overloading of the MV/LV distribution transformers. The controller is verified through simulations on an actual utility grid facing the aforementioned challenges, demonstrating the effectiveness and practicality of the proposed DRL-based smart charging approach.**

***Index Terms*—Deep deterministic policy gradient, deep reinforcement learning, demand response, distribution system, electric vehicle, machine learning, photovoltaic.**

NOMENCLATURE

Indices

e	Episode index
i	Transformer index
j	EV index
k	Future discrete time period
t	Discrete time period

Parameters

A	Exponential Zone amplitude [V]
B	Exponential Inverse time constant [Ah^{-1}]
$BEV_{i,j}$	Capacity of EV j connected to transformer i [kWh]
E^{Ah}	Individual battery cell capacity [Ah]
ε	Scaling factor
K	Polarization voltage [V]
I_{rr}	Solar irradiance [kW/m^2]
LF_{ref}	Reference load factor [p.u.]
n_p	Number of parallel branches of series-connected battery cells in the battery pack.
N	Number of discrete time steps k in which the agent interacts with the environment
NE_i	Number of EVs connected at transformer i
n_s	Number of series-connected battery cells in the battery pack.
η_c	ac-dc converter efficiency [p.u.]
η_{PV}	PV converter efficiency [p.u.]
ξ	Acceptable loading at haad-of-feeder level [p.u.]
p	probability value
P_i^{ag}	Aggregated EV demand at transformer i [p.u.]
P_{mpp}	Maximum PV array power [kW]
P^{head}	Head of the feeder active power [pu]
ϕ	Penalty reward term coefficient
φ	Reward function weights
R_{bat}	Battery cell internal resistance [Ω]
SOC^{cut}	Battery SOC at cut-off voltage [p.u.]
σ_e	Standard deviation
$\bar{S}$	Transformer's capacity [kVA]
T	Number of future discrete time periods
τ	Target network update parameter
V^{cut}	Battery cut-off voltage [V]
V_{bat}^o	Battery voltage constant [V]
θ	Policy parameters
θ^Q	Critic Network parameter vector
θ^μ	Active Network parameter vector
$\theta^{*\mu}$	Perturbed Active Network parameter vector
γ	Discount rate

Variables

a	Action
$\alpha_{i,t}$	Fraction of maximum EV demand at transformer i that can charge at time t
$\beta_{i,1,t}$	ML-S control output
$C^{\max}$	Maximum EV charging rate
$C_{i,j,t}$	Charging rate setpoint for EV$_{i,j,t}$ considering the ML-S$_i$ controller action.

The work presented in this paper has been funded by an NSERC-Alliance/OCE-VIP grant, Bluwave-ai Inc., and the University of Waterloo.

I. Calero, C. Cañizares, and K. Bhattacharya are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, N2L 3G1, Canada (e-mail: icalero@uwaterloo.ca; ccanizar@uwaterloo.ca; kankar@uwaterloo.ca).

M. Farrokhabadi is with with the University of Calgary, Calgary, AB, Canada (e-mail: mostafa.farrokhabadi@ucalgary.ca).

$C'_{i,j,t}$	Charging rate setpoint of $\text{EV}_{i,j,t}$ without the ML-S_i controller action
$\mathbb{E}$	Expected value
$E^{\text{ag}}_{i,t}$	Aggregated EV energy that needs to be supplied to charge all EVs connected at transformer i at time t
f	Piece-wise function mapping solar irradiance and correction factor
I_{bat}	Battery cell current [A]
J	Actor network cost function
$\tilde{J}_e$	Average actor loss for the episode e
L	Critic network cost function
$\ell_{i,t}$	Transformer's i loading level at time t [p.u.]
M	Number of episodes
$\mathcal{N}$	Noise process
$\Omega_{i,t}$	Reward function auxiliary variable
P_i	Transformer's i active power [kW]
P^{EV}_{dc}	Active power sent to the battery pack [W]
P^{EV}	AC charger power [W]
$\hat{P}^{\text{EV}}_{i,j,t}$	Charging power of $\text{EV}_{i,j}$ at time t in per-unit of maximum charging capacity at transformer i
π	Stochastic policy
$\Psi_{i,t}$	Uniformly distributed random variable
Q	Critic network
r	Reward [p.u.]
R_t	Cumulative future reward discounted at time t [p.u.]
s	State measurement [p.u.]
SOC	State of Charge [p.u.]
$t^{\text{arr}}_{i,j}$	$\text{EV}_{i,j}$ Arrival time [min]
$\tilde{t}^{\text{dep}}_{i,j}$	Average departure time [min]
$t^{\text{c}}_{i,j}$	Time $\text{EV}_{i,j}$ has been connected since its arrival [min]
$\tau_{i,t}$	Estimated remaining time before departure [min]
μ	Actor network
V	Charger voltage [V]
V^{oc}_{bat}	Battery cell open circuit voltage [V]
$v_{i,t}$	Transformer's i secondary voltage at time t
$\chi_{i,t}$	Reward function auxiliary variable
y	Estimated Q value
$P^{\text{EV}'}_{i,j,t}$	Charging power setpoint for $\text{EV}_{i,j,t}$ [kW]

Sets

$\mathcal{A}$	Set of discrete actions
$\mathcal{E}$	Environment
$\mathcal{R}$	Reply buffer
$\mathcal{B}$	Mini batch of transitions

I. Introduction

In recent years, motivated by environmental concerns and decreasing costs, Distributed Energy Resources (DERs) has received special attention from households, industry, and commercial outfits. In this context, Photovoltaic (PV) generation has become a viable option for energy self-sufficiency and reducing the increasing energy cost, and is thus widely adopted by these consumers [1], [2]. On the other hand, driven also by competitive electricity prices, fossil-fuel based transportation and heating demand is rapidly being electrified. For example, in Canada, the national fleet of Electric Vehicles (EVs) was 90,100 units in 2018 [3], and 100% of new car sales is expected to be electric by 2040 [4].

Rapid adoption of EVs poses challenges to utilities, such as: increased system peak load; overloading of cables, wires, and transformers; increased circuit losses; and voltage drops below acceptable limits [5]. These problems can be partially addressed by applying Time-of-Use (ToU) tariffs on customers, which penalizes energy consumption at peak hours. More effectively, the operation of the EVs can be optimally coordinated to achieve a particular objective defined by the utility [6], [7]. Thus, smart chargers are necessary to control the charging rates and schedule of EVs, as well as communicate both ways with the utility; however, this requires that EV owners agree to participate in a suitable Demand Response (DR) program that allows the utility or grid operator to monitor and control the charging of their EVs. With such control schemes in place, the EV penetration level can be increased without compromising the distribution grid reliability, and possibly help defer investments in grid infrastructure and upgrades [8].

Coordinating the operation of EVs and PVs connected to Low Voltage (LV) networks for DR applications typically involves solving a multistage optimization problem. In this context, a centralized entity at the utility level controls some Medium Voltage (MV) circuit components, while an aggregator at the distribution transformer level dispatches the DERs connected to the LV grids [9]. The results, however, are highly influenced by the methods used to predict the EV connection and disconnection times, PV generation, and load profiles, which are inputs to the problem. In most existing works (e.g., [9]–[11]), the uncertainty associated with these predictions is tackled using Model Predictive Control (MPC), Stochastic Optimization (SO), or similar approaches based on probability distribution functions (pdfs), thus ultimately relying on expected values obtained from these pdfs. However, practical implementations and the optimality of these controllers are constrained by:

- Accurate prediction of the required controller input data such as load profile, EV connection, and PV power injections;
- Detailed grid models embedded in EV charging management systems;
- Computational burden that limits the operation of existing optimization-based charging controls in real-time.

To overcome the aforementioned issues, data-driven Machine Learning (ML) based controllers have been proposed to coordinate EVs in distribution grids, which avoids the need for detailed electrical grid modeling, thus considerably reducing the computational burden for actual controller implementations. For example, in [12], a Reinforcement Learning (RL) approach is proposed to learn a cost-effective day-ahead consumption plan for charging EVs, based on available charging parameters, aiming at minimizing the energy purchase risk. However, the described intelligent control focuses on

collective EV fleet behaviour only, rather than individual EVs; thus, charging rates are specified based on a heuristic approach instead. On the other hand, researchers in [13] use Q-learning, which is an RL algorithm, in a management system for an EV charging station that seeks to maximize its revenue. The station, which is partially supplied by a renewable based generator, allows the connection of multiple EVs. The main limitation of this approach is that it categorizes EVs into two types, which have pre-defined charging strategies, since in Q-Learning, the smart agent must choose his actions from a finite set of discrete actions.

Coordinating the charging of multiple EVs connected to the same distribution grid for DR applications requires monitoring a significant amount of data. In this context, Q-Learning and, in particular, Deep-Q Learning (DQL) addresses a key challenge for high dimensional input control of fast and effective mapping of inputs to outputs. However, DQL has the major drawback of not being able to efficiently handle multi-dimensional continuous outputs. Since the ultimate goal of a centralized EV controller is to define the charging power and when to charge individual EVs, the number of outputs of a controller must be, at least, the number of EVs it oversees, and these outputs must be continuous to allow the EVs to charge at different rates. To address these limitations, the Deep Deterministic Policy Gradient (DDPG) algorithm presented in [14] is used here. DDPG is a model-free, off-policy actor-critic algorithm that uses DNNs as approximator that can learn high-dimensional continuous spaces, which makes it a suitable candidate for a central EV controller. In particular, it is based on a deterministic policy (actor) that propagates its inputs through a DNN to its high-dimensional-continuous output, thus facilitating controlling complex systems in real-time with relative ease, without the computational challenges of previously proposed optimization-based approaches such as [9] and [10] that constraint their real-time applications for EV charging control.

DDPG has been proven effective in many control applications involving physical systems, communications, robotics, to name a few. In recent years, it has received special attention in problems involving EV control. For example, [15] proposes a dynamic pricing strategy for EV aggregators based on the DDPG algorithm. Similarly, [16] proposes a DDPG algorithm-based EV charging pricing scheme for an aggregator, which addresses the limitation of existing formulations that consider fixed, discrete charging/discharging EV powers. However, these works focus on EV participation in electricity markets and their response to market signals, as opposed to solving local EV-related issues at the distribution level, which require monitoring the grid-related variables. On the other hand, [17] and [18] concentrate on the perspective of a single EV owner, proposing management systems based on DDPG, seeking to minimize the charging cost, with [18] also seeking to maintain the EV's battery energy at a desired level at the end of a charging event. The authors in [17] demonstrate that their proposed approach can achieve more than 70% cost reduction compared to other existing methods. The proposed method in [18], in addition to significant cost reduction, also shows higher user satisfaction based on their EVs' charging requirements. A similar approach is discussed in [19], which also considers stationary batteries and local PV generation. In [20], DDPG is used to maximize the EV charging from local PV generation, while also maximizing the EV's SOC at departure. In all these works, the issue of coordinating multiple EVs connected to the same grid is not addressed; instead, the intelligent DDPG agents are trained to learn optimal strategies focused on pursuing individual EV charging objectives.

The coordinated charging of multiple EVs based on Deep RL (DRL) has been discussed in few recent works. For example, [21] proposes a bi-level DRL-based coordinated charging strategy for EVs aiming to minimize the charging costs. In the first level, a Constrained-DRL (CDRL) approach finds the total charging energy of all EVs, while in the second level, a rule-based heuristic policy distributes the total charging energy among the EVs. Proximal Policy Optimization (PPC) and Soft-Actor Critic (SAC) algorithms, which can handle a high dimensional space, are used to solve the first level problem. Even though this work addresses the issue of handling a large action space, it only focuses on cost minimization; thus, the technical issues of connecting a large amount of EVs are not investigated, and the grid is not modeled either. In [22], a DC power flow model is incorporated as a set of constraints in a DRL-based decentralized model that seeks to minimize the energy cost of an energy hub with EVs. However, this work does not focus on controlling individual EVs, but the participation of multiple EV aggregators in the energy demand/supply of the hub. The authors in [23] propose a DRL-based Energy Management System (EMS) for a distribution grid with DERs, including EVs, that minimizes the operating cost of the grid. The proposed EMS, which is formulated as a traditional multi-objective optimization problem, uses the concept of Q function in a DRL algorithm to make decisions on the controlling variables. However, the proposed algorithm does not model the grid and the EVs are simply modeled as a constant-power load. In [24], Vehicle-to-the Grid (V2G) mode is assumed in a DRL-based formulation that penalizes EVs not reaching a desired SOC target at departure, in a problem that seeks to minimize carbon emissions. The problem is described as a Markov Decision Process (MDP), which limits the EV charging process, as MDP can only handle discrete action spaces.

Some works have proposed DRL-based frameworks for voltage regulation in Active Distribution Networks (ADN), not focusing in EV control and representing the grid through a balanced DistFlow model [25]–[27], balanced radial ac power flow [28], or balanced radial ac power flow model [29]. A few works pertaining EV control model the gird; for example, [30] proposes a DRL-based two-stage framework for voltage control in distribution grids, coordinating voltage regulation devices and EVs connected to charging stations. In particular, a DDPG-based method controls the charging of EVs operating in reactive power control mode while charging at a fixed active power level, as well as EVs in V2G mode exchanging active and reactive power with the grid. However, this work does not take into account other grid-related issues such as transformer overloading, which can be significant in grids with high penetration of EVs.

From the literature review, it can be noted that some of the state-of-the-art papers on RL-based coordinated EV control ignore power flow constraints or rely on simplistic DC power flows, while other works in RL-based voltage control of ADN use a balanced DistFlow model for radial systems. In contrast, our proposed approach is based on a real-world distribution system modeled in detail in the OpenDSS platform [31], which has been widely used in research and academia, particularly for EV integration studies [32]–[35] and validated against actual data in [36]; thus, unbalanced three-phase ac power flows are used during both the training and test phases. Furthermore, the state-of-the-art on RL-based coordinated EV charging models EV load as constant-power loads, which does not properly represent the charging behavior of EVs, particularly under a high sampling/execution rate paradigm; hence, the EV charging models proposed here are based on and validated with actual battery pack, converter, and local charger controller architectures and data. Therefore, the main contributions of this work are the following:

- Develop and demonstrate a novel two-level ML-based coordinated EV control trained and validated using a real-world test system and associated data.
- Consider power flow constraints during training and testing utilizing a flexible reward function, state variables, and an unbalanced ac power flow model for RL-based coordinated charging of EVs.
- Use of a realistic EV and data-validated model for unbalanced power flow modeling in distribution systems, which includes the battery pack, converter, and local charger controller.

The rest of the paper is organized as follows: Section II presents an overview of EV charging and the DDPG algorithm. In Section III, the proposed controller structure and its representation in the DDPG framework are presented. Section IV describes and discusses the test system and the controller training, and demonstrates its effectiveness to facilitate a large EV penetration in an existing grid through detailed simulations. The main conclusions of the presented work and possible future improvements are discussed in Section V.

II. Background Review

A. EV Charging Modeling and Control

Three standard charging levels are commercially available for EVs; Level 1 and Level 2 allow up to 1.8 kW and 19.2 kW charging power, respectively, while Level 3 charger power is in the range of 50 kW to 150 kW [6]. The charger can be off or on board, with the latter being more common for Level 1 and Level 2 chargers, while the former is used for Level 3 [37], [38]. In North America, Level 1 and Level 2 chargers are connected to single-phase circuits at 120 V and 240 V, respectively, and are usually installed for residential EV charging, which is the focus of this paper.

A typical charger comprises an ac-dc converter interfacing the ac grid side and a dc link, and a dc-dc converter that connects the dc link and a battery pack [37]. The ac-dc converter regulates the dc link voltage and EV power factor, whereas the dc-dc converter controls the charging power [39]. In most cases, a combination of Constant Current (CC) and Constant Voltage (CV) strategies are used to charge EVs. In CC, the charger regulates the battery pack voltage to keep the battery current constant. On the other hand, in CV, the current is regulated to keep the voltage constant. In a combined CCCV strategy, CC is applied until a cut-off voltage is reached; afterwards, it charges in CV mode until the battery is fully charged [38]. This prevents the battery pack from being overcharged when its maximum SOC is reached, thus avoiding shortening the battery life [38].

The model of the EV charger used in this work, which is based on a CCCV strategy, is depicted in Fig. 1, with the ac-dc and dc-dc converter represented in a single block. Note that since the time resolution of the proposed controller is in the min/hour range, the model neglects the dynamics of the converters and their controllers, which are in the range of ms. In this figure, the battery pack comprises an array of individual battery cells, connected in series to achieve the desired voltage at the low voltage (LV) terminal of the dc-dc converter, and in parallel to obtain the required EV energy rating. Thus, n_p parallel branches of n_s series-connected battery cells form the battery pack. The battery cells are modeled as a controllable voltage sources in series with resistors, which represent the battery cell open circuit voltage $V_{\text{bat}}^{\text{oc}}$ and internal resistant R_{bat}, respectively [40]. Hence, the EV charging relation can be modeled as follows:

$$V_{\text{bat}}^{\text{oc}} = V_{\text{bat}}^{\text{o}} - \frac{K}{SOC} + Ae^{-BE^{\text{Ah}}(1-SOC)} \tag{1}$$

$$V = \min\left\{n_{\text{s}}(V_{\text{bat}}^{\text{oc}} + R_{\text{bat}}I_{\text{bat}}), V^{\text{cut}}\right\} \tag{2}$$

$$I_{\text{bat}} = \begin{cases} CE^{\text{Ah}} & \text{if } SOC \leq SOC^{\text{cut}} \\ \dfrac{V^{\text{cut}} - n_{\text{s}}V_{\text{oc}}^{\text{bat}}}{n_{\text{s}}R_{\text{bat}}} & \text{if } SOC > SOC^{\text{cut}} \end{cases} \tag{3}$$

$$P_{\text{dc}}^{\text{EV}} = n_{\text{p}}VI_{\text{bat}} \tag{4}$$

$$P^{\text{EV}} = \frac{1}{\eta_{\text{c}}}P_{dc}^{\text{EV}} \tag{5}$$

where $V_{\text{bat}}^{\text{o}}$ is the battery voltage constant; A is the exponential zone amplitude in V; B is the exponential inverse time constant in Ah^{-1}; E^{Ah} is the individual battery cell capacity in Ah; SOC is the EV's battery pack State of Charge (SOC) in p.u.; I_{bat} is the battery cell current in A; K is the polarization voltage in V; C is the EV charging rate in p.u.; SOC^{cut} is the battery SOC when the charger voltage V of the battery reaches the cut-off voltage V^{cut}; $P_{\text{dc}}^{\text{EV}}$ is the active power sent to the battery pack; P^{EV} is the ac charger power; and η_{c} is the ac-dc converter efficiency.

B. Deep Deterministic Policy Gradient (DDPG) Algorithm

RL, which is an area of Machine Learning (ML), describes an intelligent agent that performs certain actions $a_t \in \mathbb{R}$ on an environment $\mathcal{E}$ in discrete time steps t, and can learn to modify its behaviour by observing the consequences of these actions, in response to state measurements $s_t \in \mathbb{R}^n$ of the environment and rewards associated with action-state pairs $r_t \in \mathbb{R}$ [41]. The agent behaviour is determined by a policy $\pi_\theta(s, a)$ with parameters θ, that maps state-action pairs to

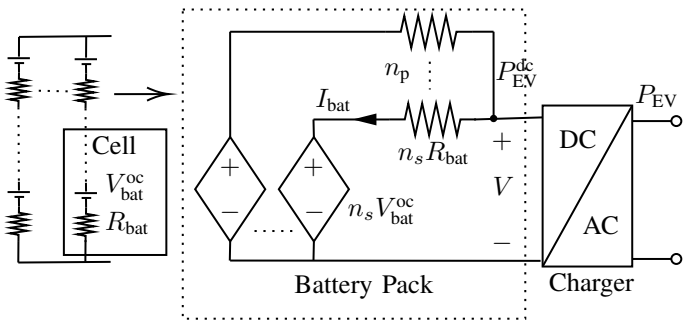


Fig. 1. EV charger model.

a probability distribution over a discrete action space, i.e., $\pi_\theta(a, s) = \Pr(a = a_t | s = s_t)$, where a belongs to a finite set of discrete actions $\mathcal{A}$.

In general, the objective of an RL agent is to directly or indirectly learn a policy that maximizes its future rewards. This can be achieved through model-based, policy-based, or value-based RL algorithms. Model-based algorithms learn the dynamics of the environment and use that information to predict the optimal strategy that the agent should follow to maximize its future rewards. Policy-based algorithms learn the policy itself, and value-based algorithms learn functions that estimate how good it is for the agent to take certain actions while being in a given state. In DRL, the policy or value functions are modeled using Neural Networks (NNs).

Q-Learning is a value-based algorithm that estimates a function, known as Q function, describing the expected discounted future rewards after taking an action a_t in state s_t following policy π, as follows [14]:

$$R_t = \sum_{k=t}^{T} \gamma^{k-t} r(s_k, a_k) \tag{6}$$

$$Q^\pi(s_t, a_t | \theta^Q) = \mathbb{E}_{r_{k\geq t}, s_{k>t} \sim \mathcal{E}, a_{k>t} \sim \pi} \left[R_t | s_t, a_t \right] \tag{7}$$

where R_t is the cumulative future rewards discounted at time t; T is number of future discrete time periods k; $\gamma < 1$ is the discounting rate; $r(s_k, a_k) \in \mathbb{R}$ is a reward function; $Q^\pi(s_t, a_t | \theta^Q)$ is the Q function with parameters θ^Q; and $\mathbb{E}$ denotes the expected value with future ($k \geq t$) states s_k and actions a_k belonging to the stochastic environment $\mathcal{E}$ and stochastic policy π, respectively. Note that (7) can be expressed in a recursive way, know as Bellman equation, as follows [14]:

$$Q^\pi(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim \mathcal{E}} \left[r(s_t, a_t) + \gamma \mathbb{E}_{a_{t+1} \sim \pi} \left[Q^\pi(s_{t+1}, a_{t+1}) \right] \right] \tag{8}$$

Traditional Deep Q-Learning has the limitation of dealing with a single-dimension discrete action space only. Hence, the authors in [14] address this issue by proposing the DDPG algorithm, which considers a deterministic policy $\mu : S \rightarrow A$, such that $\mu(s_t) = a_t \in \mathbb{R}^n$, resulting in the following Q function:

$$Q^\mu(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim \mathcal{E}} \left[r(s_t, a_t) + \gamma Q^\mu(s_{t+1}, a_{t+1}) \right] \tag{9}$$

Note that since $\mu(s_t) = a_t$ is deterministic, the inner expectation $\mathbb{E}_{a_{t+1} \sim \pi}$ is removed in (9), thus $Q^\mu(s_t, a_t)$ is only stochastic on the environment $\mathcal{E}$; this also implies that Q^μ can be learned off-policy, i.e., the exploration of the environment does not need to follow the policy μ. The DDPG algorithm aims to find the parameters θ^Q of the Q function approximator $Q(s, a | \theta^Q)$, known as critic network, by minimizing its error with respect to the Bellman equation (9). The deterministic policy $\mu(s_t | \theta^\mu)$, known as actor network, is learned by maximizing the expected Q value of the state-action pair (s, a), as follows:

$$\min_{\theta^Q} L(\theta^Q) = \frac{1}{N} \sum_{k}^{N} \left(y_k - Q(s_k, a_k | \theta^Q) \right)^2 \tag{10}$$

$$y_k = r_k + \gamma Q \left(s_{k+1}, \mu(s_{k+1} | \theta^\mu) | \theta^Q \right) \tag{11}$$

$$\max_{\theta^\mu} J(\theta^\mu) = \frac{1}{N} \sum_{k}^{N} Q(s_k, \mu(s_k | \theta^\mu)) \tag{12}$$

where L and J are the costs functions associated with the learning process of the critic and actor networks Q and μ, respectively; N is the number of discrete time steps k during which the agent interacts with the environment; y_k is the estimated Q value based on the Bellman equation (9); and θ^Q and θ^μ are the weights of the critic and actor networks, respectively. Note that the maximization of J aims to obtain a policy that also maximizes the rewards the agent obtains through time, while the minimization of L intends to make the agent learn the Q function. Also note that (11) assumes that $Q(s_{t+1})$ and $\mu(s_{t+1})$ are known to calculate $Q(s_t)$; however, this causes instabilities during training [14], as these functions are being learned. Hence, the DDPG algorithm proposes the use of target networks Q' and μ' that are identical to Q and μ, but their weights $\theta^{Q'}$ and $\theta^{\mu'}$ are updated less often than θ^Q and θ^μ.

Algorithm 1 summarizes the DDPG algorithm, where $\mathcal{R}$ is a memory buffer of fixed size where transitions (s_t, a_t, r_t, s_{t+1}) are stored at every step t. In this algorithm, M is the number of episodes; $\mathcal{N}$ is a noise process added for environment exploration; the parameter τ controls the update of the target networks.

III. EV Charger Control

The overall objective of the proposed EV charging controller is to enable larger penetration of EVs in distribution systems. This is possible by coordinating the charging of individual EVs to achieve a common goal such as preventing distribution transformers overloading while charging, which is a main concern in practice for distribution system operators. The nonlinearities of distribution grids and their components, and the stochastic behaviour of the residential demand, including EVs, and PV injections, make smart controllers ideal candidates for EV charging control. To this effect, actor networks $\mu(s | \theta^\mu)$ of DDPG agents are considered in this work as intelligent controllers, whose structure and implementation in the DDPG framework are discussed in this section.

Algorithm 1 DDPG algorithm [14]

Randomly initialize Q and μ with weights θ^Q and θ^μ
Initialize Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$
Initialize reply buffer $\mathcal{R}$
for episode=1 **to** M **do**
  Initialize a random process $\mathcal{N}$ for action exploration
  Receive initial observation state s_1
  for $t = 1$ **to** T **do**
    Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$
    Execute action a_t and observe r_t and s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{R}$
    Sample a random minibatch of N transitions (s_k, a_k, r_k, s_{k+1}) from $\mathcal{R}$
    Set $y_k = r_k + \gamma Q'\left(s_{k+1}, \mu'(s_{k+1}|\theta^{\mu'})|\theta^{Q'}\right)$
    Update critic network by minimizing the loss:
      $L = \frac{1}{N}\sum_k \left(y_k - Q(s_k, a_k|\theta^Q)\right)^2$
    Update the actor policy using the sampled policy gradient:
      $\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_k \nabla_{\theta^\mu} Q(s, a|\theta^Q)|_{s=s_k, a=\mu(s_k|\theta^\mu)}$
    Update the target networks:
      $\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$
      $\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$
  end for
end for

A. Unbalanced Distribution System Model and EV Charger Model

The environment used for training and testing the DDPG based controller discussed in this section comprises an actual unbalanced North American distribution system to which EVs, PVs, and residential customer loads are connected. To model the test distribution system, EPRI's Open-Sourced Distribution System Simulator (OpenDSS) was used [31]. This program allows simulating unbalanced multi-phase power flows for any grid configuration, including overhead or underground three- or four-wire feeders, single-phase and two phase networks, and other feeder configurations. It can also be customized to interact with other platforms, such as Python. The model of a generic customer with EV and PV is depicted in Fig. 2, based on [42], [43].

The PV system is connected to one of the phases and the neutral of a centre tapped single-phase distribution transformer, at 120 V. Assuming the PV system has maximum power tracking control, the active power at the ac side of the converter (P^{PV}) can be calculated as follows [42]:

$$P^{\text{PV}} = \eta_{\text{PV}} P_{\text{mpp}} Irr\, f(Temp) \tag{13}$$

where η_{PV} is the converter efficiency; P_{mpp} is the maximum power of the PV array in kW; Irr is the solar irradiance onto the PV array's surface in kW/m^2; and f is a piece-wise function that maps the temperature $Temp$ of the solar array in ^{o}C to a correction factor in per unit of P_{mpp}.

The EV charging model described in Section II-A, receives the EV's charging rate C and SOC as input and yields P^{EV} as output. A storage element in OpenDSS can be used to model the SOC, as illustrated in Fig. 2, where the the effective power at the dc side of the charger is integrated in a storage device, considering the converter efficiency η_{c}, fixed idling losses P_{idl}, and charging efficiency η_{ch}, with the latter representing the battery back losses. It is assumed that the EVs have Level 2 charging capacity, and thus are connected at 240 V. The customer demand P_{L} is modeled as a constant power load, evenly distributed between the two accessible phases at the transformer terminals.

B. Controller Structure

The proposed controller structure comprises two levels, as depicted in Fig. 3; the primary level (ML-P) coordinates the operation of secondary controllers (ML-S$_i$), each associated with a MV/LV transformer to achieve the desired DR objective at the head of the feeder. Specifically, the ML-P controller aims to minimize the active power at the head of the feeder P^{head} and improve the feeder's Load Factor ($LF = \left[\frac{1}{24}\sum_h^{24}(P_h^{\text{head}})\right]/\max(P_h^{\text{head}})$), by setting the maximum aggregated EV demand P_i^{ag} that each transformer i can handle [9]. This information is passed to the ML-S$_i$ controllers as setpoints for their aggregated EV load. On the other hand, the secondary controllers seek to minimize their associated transformer overload due the EV charging, minimize transformer's voltage deviations beyond acceptable limits, maximize the EVs' SOC at departure, and follow the setpoints P_i^{ag} determined by the ML-P controller. Note that other possible states, such as flows through feeder sections (main or laterals) are not considered, as existing measurement equipment is only available at substation level or step-down distribution system transformers through SCADA systems or Distribution Transformer Monitoring (DTM) systems. For the controller training and testing, OpenDSS is used to model a detailed unbalanced distribution system, as illustrated in Figs. 4 and 5, which is impacted by the controller actions on the EV chargers, and from where state measurements are obtained.

C. Controller Implementation in the DDPG Framework

1) ML-P Controller: The proposed ML-P controller structure is depicted in Fig. 4. The state space is formed by measurements of the following: loading $\ell_t = \{\ell_{1,t}, ..., \ell_{NT,t}\} \in \mathbb{R}^{NT}$ of all NT transformers connected at the feeder at time t in p.u., where $\ell_{i,t} = P_i/\bar{S}_i$, with P_i and $\bar{S}_i$ representing the active power and the kVA capacity of the distribution transformer i, respectively; $P^{\text{head}} \in \mathbb{R}$ in p.u. with respect to the substation (S/S) transformer maximum capacity; and the aggregated EV energy that needs to be supplied to fully charge all EVs connected at each transformer i at time t, i.e., $E_t^{\text{ag}} = \{E_{1,t}^{\text{ag}}, ..., E_{NT,t}^{\text{ag}}\} \in \mathbb{R}^{NT}$. $E_{i,t}^{\text{ag}}$ is defined as follows:

$$E_{i,t}^{\text{ag}} = \frac{1}{\sum_j BEV_{i,j}} \sum_j^{NE_i} (1 - SOC_{i,j,t})\, BEV_{i,j} \quad \forall j = 1, .., NE_i \tag{14}$$

where $BEV_{i,j}$ is the capacity in kWh of EV$_{i,j}$, i.e., the EV j connected at transformer i; $SOC_{i,j,t}$ is the SOC of EV$_{i,j}$; and NE_i is the number of EVs connected at transformer i. Note

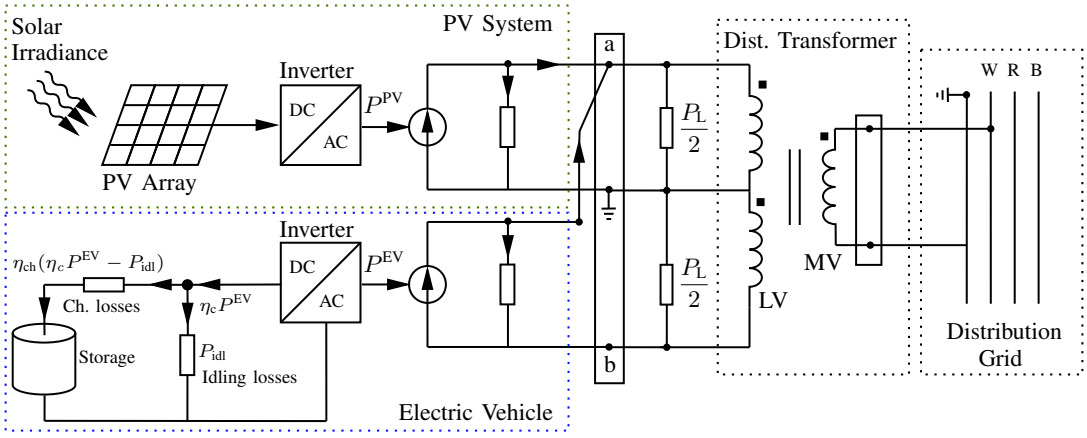


Fig. 2. Model of a generic customer in OpenDSS.

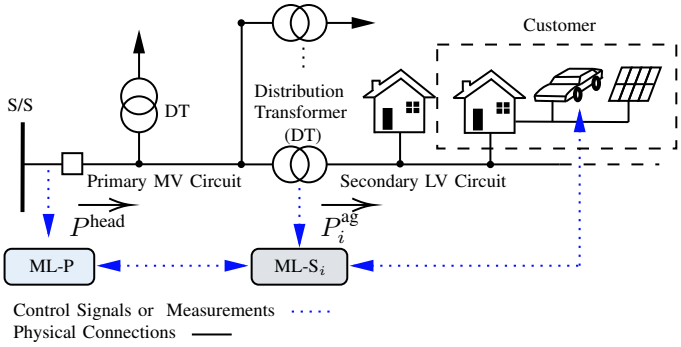


Fig. 3. Proposed controller structure

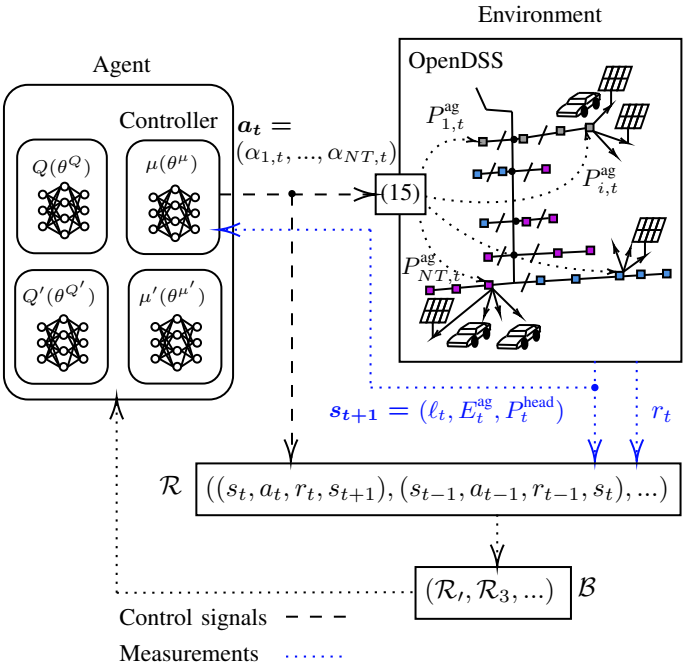


Fig. 4. ML-P controller described as a DDPG agent.

that $E_{i,t}^{ag}$ is calculated by each ML-S$_i$ controller based on its local information and then passed to the ML-P controller, and hence, $s_t = (\ell_t, E_t^{ag}, P^{head}) \in \mathbb{R}^{2NT+1}$. The proposed state space intends to provide the agent with enough information to capture the system loading and EV energy to be supplied.

$a_t = (\alpha_{1,t}, ..., \alpha_{NT,t}) \in \mathbb{R}^{NT}$ is the action space, where $\alpha_{i,t}$ is defined as the fraction of the total EV capacity connected at transformers i that can charge at time t. Hence, the P_i^{ag} setpoints in kW can be calculated as follows:

$$P_{i,t}^{ag} = \alpha_{i,t} \sum_{j}^{NE_i} BEV_{i,j} C_{i,j}^{max} \tag{15}$$

where $C_{i,j}^{max} = \min\{0.5, 19.2/BEV_{i,j}\}$ is the maximum EV charging rate, with 19.2 being the maximum power in kW that a Level 2 charger can supply.

The reward function has two objectives: penalizing the overloading of the substation (S/S) transformer and rewarding flat daily demand profiles. For the first objective, it is assumed that P^{head} reasonably represents the S/S transformer loading, with P^{head} = 1 p.u. representing the S/S transformer nominal capacity. Hence, when P^{head} is larger than an assumed acceptable loading level ξ (e.g., ξ =1.2 p.u. signifies that the transformer can be occasionally overloaded by 20%), the agent is penalized. On the other hand, when the S/S transformer is not significantly overloaded, i.e., $P^{head} < \xi$, the agent receives a reward that depends on how close P^{head} is to a desired p.u. flat demand, calculated based on a reference Load Factor $LF_{\text{ref}} \in [0, 1]$ defined by the operator. Hence, the reward function is defined as follows:

$$r_t = \begin{cases} -\phi_1 \left(P_t^{head} - \xi\right)^2 & \text{if } P_t^{head} > \xi \\ \phi_2 \left(1 - \left|LF_{\text{ref}} - P_t^{head}\right|\right) & \text{if } P_t^{head} \leq \xi \end{cases} \tag{16}$$

where $\phi_1 > 0$ and $\phi_2 > 0$ are coefficients for the penalty and reward terms, respectively.

2) ML-S Controller: The proposed structure for the ML-S$_i$ controller associated with transformer i is illustrated in Fig. 5. The state space includes the SOC and per-unit charging power measurements of all EVs connected at transformer i at time t, i.e., $SOC_{i,t} = \{SOC_{i,1,t}, ..., SOC_{i,NE_i,t}\} \in \mathbb{R}^{NE_i}$ and $\hat{P}_{i,t}^{EV} = \{\hat{P}_{i,1,t}^{EV}, ..., \hat{P}_{i,NE_i,t}^{EV}\} \in \mathbb{R}^{NE_i}$. $\hat{P}_{i,j,t}^{EV}$ is defined as

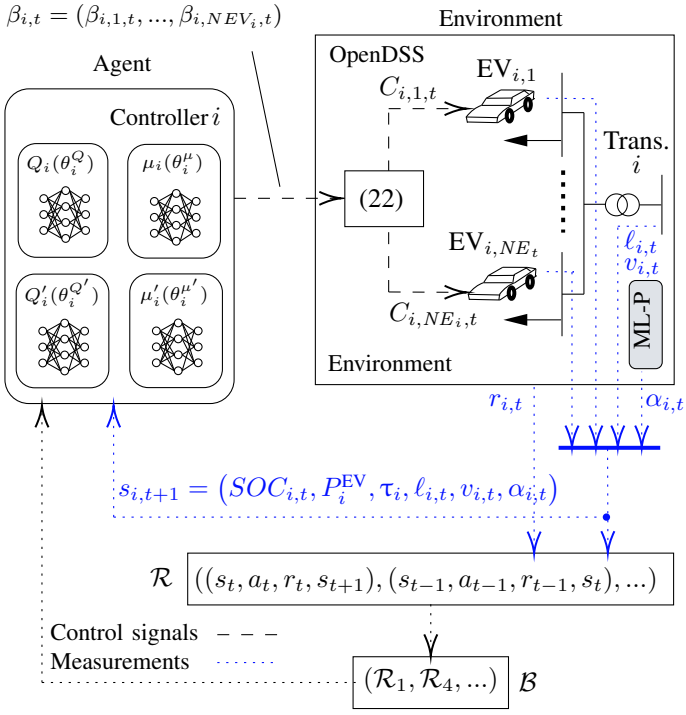


Fig. 5. ML-S controller described as a DDPG agent.

follows:

$$\hat{P}^{\text{EV}}_{i,j,t} = \frac{P^{\text{EV}}_{i,j,t}}{\sum_j^{NE_i} BEV_{i,j} C^{\max}_{i,j}} \tag{17}$$

where $P^{\text{EV}}_{i,j,t}$ is the measured charging power of $\text{EV}_{i,j}$ in kW.

The state vector also considers the estimated remaining time before departure for each $\text{EV}_{i,j}$ $\tau_{i,t} = \{\tau_{i,1,t}, ..., \tau_{i,NE_i,t}\} \in \mathbb{R}^{NE_i}$. $\tau_{i,j,t}$ is defined as follows:

$$\tau_{i,j,t} = \frac{1}{1,440}\left(\left[(1,440 - t^{\text{arr}}_{i,j}) + \tilde{t}^{\text{dep}}_{i,j}\right] - t^{\text{c}}_{i,j}\right) \tag{18}$$

where $t^{\text{arr}}_{i,j}$, $\tilde{t}^{\text{dep}}_{i,j}$, and $t^{\text{c}}_{i,j}$ are the arrival time, average departure time, and the time $\text{EV}_{i,j}$ has been connected since its arrival, in minutes, respectively; note that 1,440 corresponds to the number of minutes in a day. The arrival and average departure times are measured with respect to 00:00h every day. Furthermore, $\tilde{t}^{\text{dep}}_{i,j}$ is a simple average calculated for each $\text{EV}_{i,j}$ every day, based on W historical data measurements stored in the controller memory of their corresponding actual departure times $t^{\text{dep}}_{i,j,w}$, as follows:

$$\tilde{t}^{\text{dep}}_{i,j} = \frac{1}{W}\sum_w^W t^{\text{dep}}_{i,j,w} \tag{19}$$

Finally, the state vector also contains the local transformer loading $\ell_{i,t} \in \mathbb{R}$, its per-unit secondary-side voltage $v_{i,t} \in \mathbb{R}$, and the ML-P controlling signal $\alpha_{i,t} \in \mathbb{R}$, and hence, $s_{i,t} = \left(SOC_{i,t}, \bar{P}^{\text{EV}}_{i,t}, \tau_{i,t}, \ell_{i,t}, v_{i,t}, \alpha_{i,t}\right) \in \mathbb{R}^{3NE_i+3}$.

The aggregated EV power $P^{\text{ag}}_{i,t}$ in (15) is distributed among all $\text{EV}_{i,j}$ prioritizing those with little energy available in their batteries as follows:

$$P^{\text{EV}'}_{i,j,t} = \left[\frac{(1 - SOC_{i,j,t})BEV_{i,j}}{\sum_j^{NE_i}(1 - SOC_{i,j,t})BEV_{i,j}}\right] P^{\text{ag}}_{i,t} \tag{20}$$

where $P^{\text{EV}'}_{i,j,t}$ is the charging power setpoint for $\text{EV}_{i,j,t}$ in kW. The charging power $P^{\text{EV}'}_{i,j,t}$ can be readily expressed in terms of the charging rate of $\text{EV}_{i,j,t}$ as follows:

$$C'_{i,j,t} = \frac{P^{\text{EV}'}_{i,j,t}}{C^{\max}_{i,j} BEV_{i,j}} \tag{21}$$

where $C'_{i,j,t}$ is the $\text{EV}_{i,j,t}$ charging rate setpoint without the ML-S_i controller action. Hence, the action space of ML-S_i controllers corresponds to the vector $\beta_{i,t} = \{\beta_{i,1,t}, ..., \beta_{i,NE_i,t}\} \in \mathbb{R}^{NE_i}$, with $\beta_{i,j,t}$ affecting the charging rate setpoints $C'_{i,j,t}$ as follows:

$$C_{i,j,t} = C'_{i,j,t}\,(1 + \beta_{i,j,t}) \tag{22}$$

where $C_{i,j,t}$ is the charging rate applied to $\text{EV}_{i,j,t}$ considering the ML-S_i controller action.

The reward function penalizes the agent when the local transformer overloads due to EV charging, the transformer's secondary-side voltage is outside the utility's acceptable limits $v_{\min}$ and $v_{\max}$, or when EVs charge while the transformer is already overloaded. It also penalizes EV departures without being fully charged, in proportion of the SOC at departure $SOC^{\text{dep}}_{i,j,t}$. On the other hand, rewards are received by the agent when the ML-P setpoints $P^{\text{ag}}_{i,t}$ are closely followed. Hence, the following reward function is proposed:

$$\begin{aligned} r_{i,t} = \Omega_{i,t} + \chi_{i,t} - \varphi_2 \sum_j^{NE_i} \sqrt{\left|1 - SOC^{\text{dep}}_{i,j,t}\right|} \\ + \varphi_3 \left(1 - \frac{\left|P^{\text{ag}}_{i,t} - \sum_j P^{\text{EV}}_{i,j,t}\right|}{\sum_j^{NE_i} BEV_{i,j} C^{\max}_{i,j}}\right)^2 \end{aligned} \tag{23}$$

$$\chi_{i,t} = \begin{cases} -\varphi_1\left(\ell_{i,t} - \xi_i\right)^2 & \text{if } \ell_{i,t} > \xi_i \wedge \sum_j^{NE_i} P^{\text{EV}}_{i,j,t} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

$$\Omega_{i,t} = \begin{cases} -\varphi_4\left(v_{i,t} - v_{\max}\right)^2 & \text{if } v_{i,t} > v_{\max} \\ -\varphi_5\left(v_{\min} - v_{i,t}\right)^2 & \text{if } v_{i,t} < v_{\min} \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

where $\xi_i > 1$ is the maximum transformer loading acceptable, and the coefficients $\varphi_1 > 0$, $\varphi_2 > 0$, $\varphi_3 > 0$, $\varphi_4 > 0$ and $\varphi_5 > 0$ vary the weights of the penalty and reward terms in the reward function.

It should be mentioned that the proposed charging control does not consider battery degradation, since there is no practical method to measure degradation of the battery cells of all EVs in the feeder. Thus, it is not possible to account for this issue in the proposed algorithm, as this requires measurement of environment variables to incorporate them in the states or reward functions.

D. Practical Implementation

The information flow in the proposed controller at each time interval t is depicted in Fig. 6, where it can be observed that the ML-P controller measures the active power at the head of the feeder to determine the controller input P^{head}, sending setpoints P_i^{ag} to all controllers ML-S$_i$. Each ML-S$_i$ measures the local transformer's voltage v_i and loading level to determine ℓ_i and for each EV connected to a transformer i, a ML-S$_i$ controller receives the latest measurement $SOC_{i,j}$ and the charging power P^{EV}, both used to calculate $E_{i,t}^{\text{ag}}$ in (14), and the time the EV has been connected since its last arrival t^c. When an EV departs, its charging power goes to zero, with the ML-S$_i$ recording the departure time t^{dep} and the SOC at that point SOC^{dep}, which is used as the SOC measurement until the EV returns; when an EV arrives, the ML-S$_i$ records the arrival time t^{arr} and updates the SOC. Based on the t^{dep} recorded data, the controller ML-S$_i$ calculates $\hat{t}^{\text{dep}}$, which along with t^{arr} and t^c are used to calculate the state measurement $\tau_{i,j,t}$ in (18). Note that the choice of the controller's states was constrained by commercially available monitoring equipment, since, for example, power flows through feeder segments are not typically monitored, as previously explained, and are thus not considered in the proposed controller. For example, the head of the feeder active power could be obtained from SCADA systems, while transformer and EV data require special infrastructure (e.g., [44]), to enable the flow of information from EVs to the ML-S and the ML-P controllers, as indicated in Fig. 6. Other type of grid states, such as power flows trough segments of a feeder, are not typically monitored, thus not considered in the proposed controller.

The controller training takes place offline based on historical demand and EV data based on large real-world data sets, plus an advanced and practical distribution system model, implemented in OpenDSS and integrated in the controller environment developed in Python, thus considering for power flow constraints in the training process. Once the controller is trained, it is then deployed, thus minimizing unintended operations that could result in grid issues. The controller would be maintained by being regularly retrained using up-to-date available information. Note that the proposed EV charging control approach can be readily applied to feeders of any size, since it is not significantly impacted by the system dimension like optimization-based techniques (e.g., [9] and [10]). The feeder size would only affect the topology of the ML-P critic and actor NNs by increasing the number of the input and output neurons, as well as their respective off-line training, with the grid architecture not being a design parameter, and rather being part of the environment with which the controller interacts.

E. Modified DDPG Algorithm

The original DDPG algorithm suggests the use of a time-correlated Ornstein–Uhlenbeck process to add white noise to the action space, aiming at a larger search space to avoid falling in a local minimum. An alternate option consist of adding white noise to the parameter vector of the current

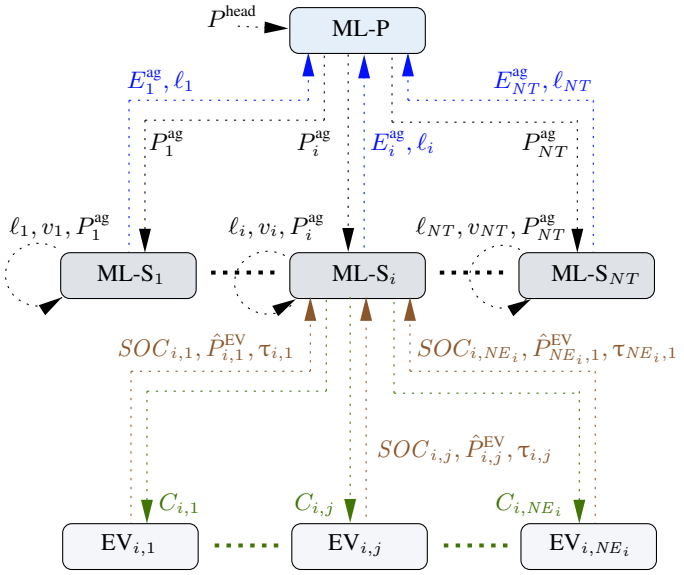


Fig. 6. Information flow in the proposed controller.

policy, which could lead to more consistent exploration, and thus, more effective learning than traditional RL with action space noise [45]. In addition, since any $\alpha_{i,t} > 0$ in the ML-P controller results in EV charging rates $C_{i,j,t} > 0$, the action space can be forced to $\alpha_{i,t} = 0$ as part of the exploration mechanism, so as to minimize frequent simultaneous EV charging. Hence, a mechanism is adopted that combines a perturbed action space and white noise addition to the parameter vector, based on [45], as follows:

$$\theta^{*\mu} = \theta^{\mu} + \mathcal{N}_t(0, \sigma_e^2) \tag{26}$$

$$\sigma_{e+1} = \begin{cases} \frac{1}{\varepsilon}\sigma_e & \text{if} \quad \tilde{J}_e > \tilde{J}_{e-1} \\ \varepsilon\sigma_e & \text{if} \quad \tilde{J}_e \leq \tilde{J}_{e-1} \end{cases} \tag{27}$$

$$\alpha_{i,t} = \begin{cases} \alpha_{i,t} & \text{if} \quad \Psi_{i,t} \leq p \\ 0 & \text{if} \quad \Psi_{i,t} > p \end{cases} \tag{28}$$

where $\theta^{*\mu}$ is the perturbed parameter vector; σ_e is the standard deviation of the normally distributed random variable $\mathcal{N}_t(0, \sigma_e^2)$ centered at 0; $\varepsilon > 1$ is a scaling factor updated at the end of each episode e; $\tilde{J}_e$ is the average actor loss for the episode e: $\tilde{J}_e = \frac{1}{T}\sum_t^T J_t$, with T being the total number of discrete time steps in the episode e.

The parameter value $p \in [0, 1]$ is a probability value; and $\Psi_{i,t} \sim U[0, 1]$ is a uniformly distributed random variable generated at time t for transformer i. Note that (27) increases the standard deviation if the performance of the agent in the current episode is worse than in the previous, expanding the search space; otherwise, the standard deviation is reduced, thus focusing more on exploiting the current policy. Observe that (28) applies to the ML-P controller only, while (26)-(27) apply to both ML-P and ML-S$_i$.

To speed up and facilitate convergence in the controller training, the two-stage procedure is proposed. Thus, in Stage 1, the ML-P controller is trained, with controllers ML-S$_i$ not affecting the EVs' charging rates, i.e., in Fig. 5, (22) is

Algorithm 2 Proposed DDPG controller training for Stage 1

Randomly initialize Q_0 and μ_0 with weights $\theta^{Q_{0_o}}$ and $\theta^{\mu_{0_o}}$
Initialize Q'_0 and μ'_0 with weights $\theta^{Q'_{0_o}} \leftarrow \theta^{Q_{0_o}}$, $\theta^{\mu'_0} \leftarrow \theta^{\mu_0}$
Randomly initialize EVs' SOC
for $t = 1$ **to** size($\mathcal{R}_0$) **do** ▷ Initialize reply buffer
  Set EV charging rates based on actions $\alpha_t = \mu_0(s_{0,t}|\theta^{\mu_{0_o}})$ and (21)
  Solve the system unbalanced power flow to obtain bus voltages and feeder flows for all phases
  Update the EV's SOC
  Observe states $s_{0,t+1}$ and calculate $r_{0,t}$ based on (24)
  Store transitions $(s_{0,t}, a_{0,t}, r_{0,t}, s_{0,t+1})$ in memory $\mathcal{R}_0$
end for
Initialize noise $\mathcal{N}_{0,t}(0, \sigma_0^2)$ parameter $\sigma_0 = \sigma_{0_o}$
for $e = 1$ **to** M **do**
  Randomly initialize EVs' SOC
  for $t = 1$ **to** T **do**
    Obtain the perturbed weights $\theta^{*\mu_{0_o}}$ based on (26)
    Select action $\alpha_t = \mu(s_{0,t}|\theta^{*\mu_{0_o}})$ based on (28)
    Obtain $P^{ag}_{i,t}(\alpha_t)$ and $C'_{i,j,t}$ based on (15) and (21), respectively
    Solve the system power flow, update the EV's SOC
    Observe states $s_{0,t}$ and calculate $r_{0,t}$ based on (24)
    Store transitions $(s_{0,t}, a_{0,t}, r_{0,t}, s_{0,t+1})$ in $\mathcal{R}_0$
    Sample a random minibatch of size($\mathcal{B}_i$) transitions $(s_{0,k}, a_{0,k}, r_{0,k}, s_{0,k+1})$ from $\mathcal{R}_0$
    Set:
    $y_{0,k} = r_{0,k} + \gamma Q'_0(s_{0,k+1}, \mu'_0(s_{0,k+1}|\theta^{\mu'_{0_o}})|\theta^{Q'_{0_o}})$
    Update critic networks by minimizing the losses:
    $$L_0 = \frac{1}{\text{size}(\mathcal{B}_0)} \sum_k \left(y_k - Q_0(s_k, a_k|\theta^{Q_{0_o}})\right)^2$$
    Update the actor policy by maximizing:
    $$J_0 = \frac{1}{\text{size}(\mathcal{B}_0)} \sum_k Q_0(s_k, \mu_0(s_k|\theta^{\mu_{0_o}})|\theta^{Q_{0_o}})$$
    Update the target networks:
    $\theta^{Q'_{0_o}} \leftarrow \tau_0 \theta^{Q_{0_o}} + (1 - \tau_0)\theta^{Q'_{0_o}}$
    $\theta^{\mu'_{0_o}} \leftarrow \tau_0 \theta^{\mu_{0_o}} + (1 - \tau_{0_o})\theta^{\mu'_{0_o}}$
  end for
  Update $\sigma_{0,e+1}$ based on (27)
end for

replaced by (21) to define the charging rates. The weights $\theta^{\mu_o}, \theta^{Q_o}, \theta^{\mu'_o}$, and $\theta^{Q'_o}$ thus obtained, are then used as initial weights for ML-P in Stage 2, where ML-P and ML-S$_i$ are trained together. The modified DDPG algorithm for Stages 1 and 2 are summarized in Algorithms 2 and 3, respectively, where index $i = 0$ is used to refer to the ML-P controller while $i = 1, .., NT$ corresponds to the NT ML-S$_i$ controllers.

IV. Results and Discussion

A. Test System

The proposed controller was tested on a portion of an actual feeder of a local utility in Ontario, Canada, as depicted in Fig. 11. The feeder comprises a four-wired wye radial primary circuit at 4.16 kV, with sections underground and overhead and single-phase laterals, which results in an ubalanced grid. There are 23 MV/LV distribution transformers, with 21 being single-phase serving 156 residential customers, and the remaining being three-phase serving 2 large commercial customers. Each customer is modeled based on the generic model illustrated in Fig. 2. The standard transformers sizes for this feeder are 25 kVA, 37.5 kVA, 50 kVA, 75 kVA, and 300 kVA (for the commercial load). PV systems and EVs were randomly added to 75% of the residential premises without considering any restrictions in terms of transformer capacity. Furthermore, the PV system sizes were randomly picked from a pool of four, i.e., 2 kW, 4 kW, 6 kW and 8 kW, while the EV battery sizes were selected from a pool of 182 actual EVs in Ontario, obtained from the Charge the North Program [46], which has actual EV data available from all over Canada. It is assumed that all EVs charge at home only. The resulting number of connected EVs and the total EV charging capacity (assuming a maximum charging rate of 0.5 or 19.2 kW) at each transformer is presented in Table I. Note that the EV charging capacity in some transformers is significantly higher than the transformer capacity, thus demonstrating the benefits of the proposed EV charging controllers to properly manage the existing distribution system assets.

To determine the system impact and issues associated with connecting EVs without any coordinated control, the feeder was first modeled using actual demand measurements at customer level for a year and a half. The loading level of all distribution transformers (step-down transformers) in the feeder and their secondary-side voltage are presented in Figs. 7 and 8, respectively. Observe that during Summer (July-August) there are several instances of transformer overloading, while voltages do not fall under the minimum limit required by the utility, i.e., $v_{\min} = 0.92$ p.u. Then, 100 scenarios for the same period were simulated assuming different penetration levels of EVs and PVs, in which these were randomly connected to customers premises and considering that 100% penetration level means that all utility customers in the feeder have an EV and PV installed. Fig. 9 summarizes the results of the 100 scenarios in a box plot of the "overloaded transformer power", calculated as the current transformer load minus the maximum transformer capacity in kW. Observe that as the penetration levels of EVs and PVs increases, the amount of power supplied in overload condition significantly increases, particularly at night when EVs are expected to be charging. On the other hand, Fig. 10 shows a box plot of the voltage level at the secondary side of all distribution transformers. Note also that even larger levels of EV and PV penetration does not result in considerable voltage violations, as also been observed in other similar studies [47], [48], due to oversized cables and wires in this system compared to the system transformers, as is typi-cally the case in distribution grids, and considering that solar PV injects only active power by operating as usual at unity power factor, the voltage input in the state space has no influence in the controller performance neither during training nor testing. Thus, voltage inputs $v_{i,t}$ and associated penalty factor $\Omega_{i,t}$ had no impact on the on the design of the controller discussed henceforth. Nevertheless, if voltage profiles would have been an issue in the real-world test system used, the proposed approach should have been able to readily consider

Algorithm 3 Proposed DDPG controller training for Stage 2

Initialize Q_0 and μ_0 with weights $\theta^{Q_{0_o}}$ and $\theta^{\mu_{0_o}}$, and randomly initialize Q_i and μ_i with weights θ^{Q_i} and θ^{μ_i} $\forall i = 1, ..., NT$
Initialize Q'_i and μ'_i with weights $\theta^{Q'_i} \leftarrow \theta^{Q_i}$, $\theta^{\mu'_i} \leftarrow \theta^{\mu_i}$ $\forall i = 0, ..., NT$
Randomly initialize EVs' SOC
for $t = 1$ **to** size($\mathcal{R}$) **do** ▷ Initialize reply buffer
  Set EV charging rates based on actions $\alpha_t = \mu_0(s_{0,t}|\theta^{\mu_0})$ and $\beta_{i,t} = \mu_i(s_{i,t}|\theta^{\mu_i})$ $\forall\ i = 1, .., NT$ and (22)
  Solve the system unbalanced power flow in OpenDSS to obtain bus voltages and feeder flows for all phases
  Update the EV's SOC and observe states $s_{i,t+1}$ $\forall\ i = 0, .., NT$
  Calculate $r_{0,t}$ and $r_{i,t}$ $\forall\ i = 1, .., NT$ based on (24) and (23), respectively
  Store transitions $(s_{i,t}, a_{i,t}, r_{i,t}, s_{i,t+1})$ in memory $\mathcal{R}_i$ $\forall\ i = 0, .., NT$
end for
Initialize noise $\mathcal{N}_{i,t}(0, \sigma_i^2)$ parameter $\sigma_i = \sigma_{i_o}$ $\forall\ i = 0, .., NT$
for $e = 1$ **to** M **do**
  Randomly initialize EVs' SOC
  for $t = 1$ **to** T **do**
    Obtain the perturbed weights $\theta^{*\mu_i}$ $\forall\ i = 0, .., NT$, based on (26)
    Select action $\alpha_t = \mu(s_{0,t}|\theta^{*\mu_0})$ based on (28) and obtain $P^{\text{ag}}_{i,t}(\alpha_t)$ based on (15)
    Select action $\beta_{i,t} = \mu_i(s_{i,t}|\theta^{*\mu_i})$ and obtain $C_{i,j,t}(\beta_{i,t})$ based on (22) $\forall\ i = 1, .., NT$ and $j = 1, ..., NE_i$
    Solve the system power flow, update the EV's SOC and observe states $s_{i,t+1}$ $\forall\ i = 0, .., NT$
    Calculate $r_{0,t}$ and $r_{i,t}$ $\forall\ i = 1, .., NT$ based on (24) and (23), respectively
    Store transitions $(s_{i,t}, a_{i,t}, r_{i,t}, s_{i,t+1})$ in memory $\mathcal{R}_i$ $\forall\ i = 0, .., NT$
    Sample a random minibatch of size($\mathcal{B}_i$) transitions $(s_{i,k}, a_{i,k}, r_{i,k}, s_{i,k+1})$ from $\mathcal{R}_i$ $\forall\ i = 0, .., NT$
    Set $y_{i,k} = r_{i,k} + \gamma Q'_i\left(s_{i,k+1}, \mu'_i(s_{i,k+1}|\theta^{\mu'_i})|\theta^{Q'_i}\right)$ $\forall\ i = 0, .., NT$
    Update critic networks by minimizing the losses:
    $$L_i = \frac{1}{\text{size}(\mathcal{B}_i)} \sum_k \left(y_k - Q_i(s_k, a_k|\theta^{Q_i})\right)^2 \ \forall\ i = 0, .., NT$$
    Update the actor policy using the sampled policy gradient:
    $$\nabla_{\theta^{\mu_i}} J_i \approx \frac{1}{\text{size}(\mathcal{B}_i)} \sum_k \nabla_{\theta^{\mu_i}} Q_i(s, a|\theta^{Q_i})|_{s=s_k, a=\mu_i(s_k|\theta^{\mu_i})} \ \forall\ i = 0, .., NT$$
    Update the target networks:
    $$\theta^{Q'_i} \leftarrow \tau_i \theta^{Q_i} + (1 - \tau_i)\theta^{Q'_i} \quad \forall\ i = 0, .., NT$$
    $$\theta^{\mu'_i} \leftarrow \tau_i \theta^{\mu_i} + (1 - \tau_i)\theta^{\mu'_i} \quad \forall\ i = 0, .., NT$$
  end for
  Update $\sigma_{i,e+1}$ $\forall\ i = 0, .., NT$ based on (27)
end for

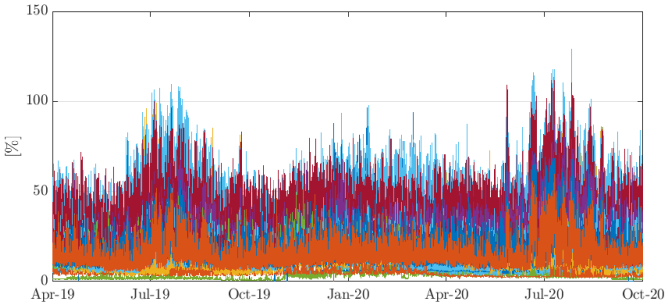


Fig. 7. Distribution (step-down) transformer loading without EVs connected.

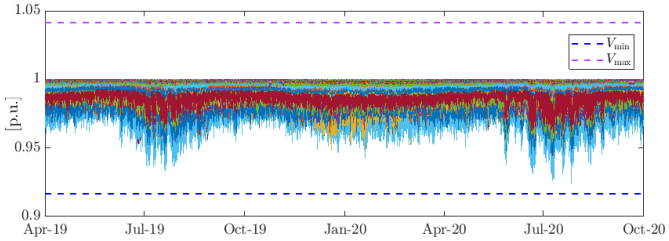


Fig. 8. Voltage at secondary side of distribution transformers without EVs connected.

them in the design of the controller, impacting the weights in the critic and actor networks as well as their size.

Note that by using real-world models to verify the proposed method, the controller developed can be readily applied to the field, given the required hardware and software infrastructure is in place. Furthermore, the proposed novel data-driven two stage model is scalable and training-efficient. Thus, it can be easily transferred from one feeder to another; it requires minimum reformulation compared to conventional methods, making it suitable for the fast-paced reconfiguration of modern distribution systems.

1) Data: The controller was trained using data for July 2020, which was the month with the highest peak demand $P_{\max}$ =750 kW without EVs in that year for the test system. Since the S/S transformer size was not available for this study, P^{head} was expressed in per unit of $P_{\max}$. The demand data available, which does not include either PV or EV injections,

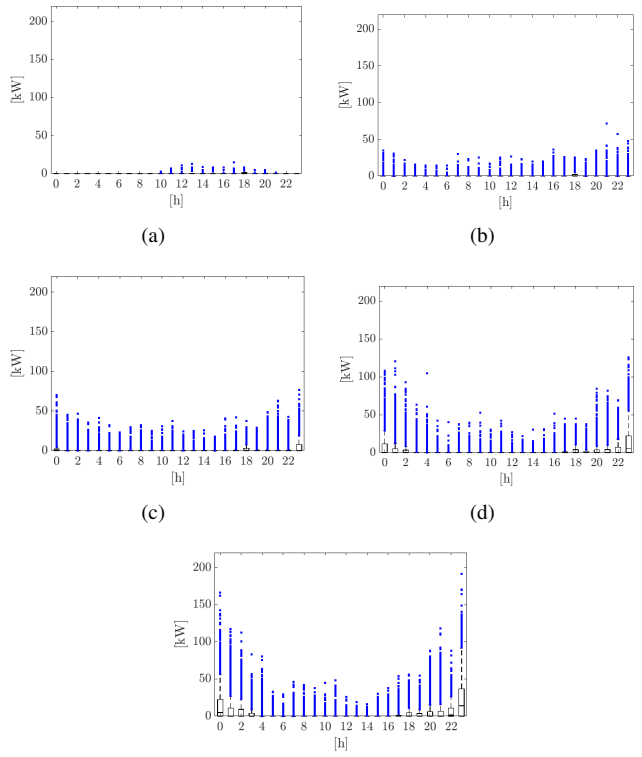


Fig. 9. Overloaded power at distribution transformers for a) 0%, b) 25%, c) 50%, d) 75%, and e) 100% EV and PV penetration levels.

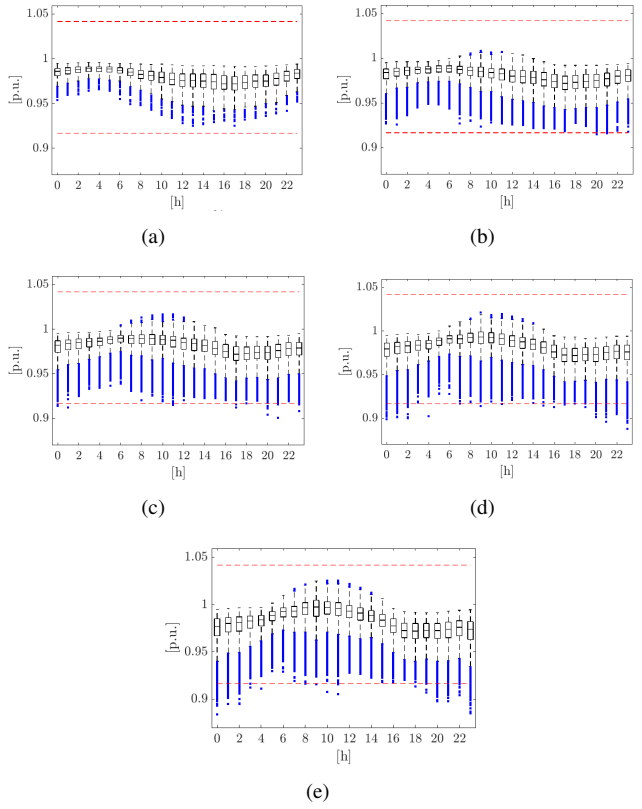


Fig. 10. Secondary side voltage of distribution transformers for a) 0%, b) 25%, c) 50%, d) 75%, and e) 100% EV and PV penetration levels.

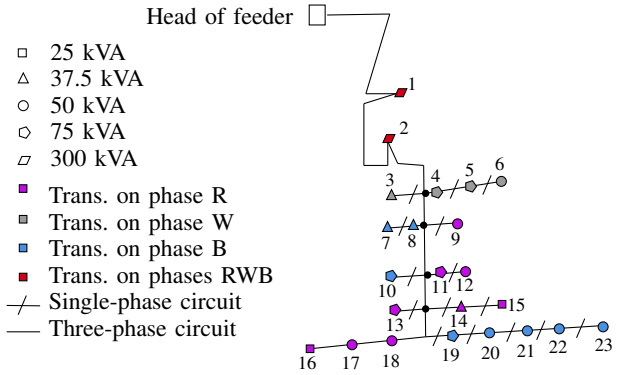


Fig. 11. Test feeder with numbered transformers.

TABLE I
NUMBER AND CAPACITY OF EVS CONNECTED AT EACH TRANSFORMER.

	Transformer Number							
	1	2	3	4	5	6	7	8
Number of EVs	0	0	6	8	3	3	3	7
Demand (kW)	0	0	94	117	32	32	47	80
	Transformer Number							
	9	10	11	12	13	14	15	16
Number of EVs	6	5	5	8	7	6	4	3
Demand (kW)	96	68	62	79	115	55	59	38
	Transformer Number							
	17	18	19	20	21	22	23	
Number of EVs	2	8	7	2	9	7	5	
Demand (kW)	39	119	90	10	123	116	71	

corresponds to hourly measurements of the active power at each premise; therefore, a 0.9 lagging power factor was assumed to arrive at an estimate of the reactive power demand at a bus.

The solar irradiance and array temperature data used as input to the PV system model was obtained from [49] for the feeder location, assuming a power factor of 1. The EV data comprises battery sizes of actual EVs in Ontario homes and charging events, from where statistical information was extracted to create synthetic data, i.e., standard (std) deviation σ_{SOC_o} and average $\widetilde{SOC}_o$ values of the initial SOC. Since arrival and departure times were not available, average $\tilde{t}^{\text{arr}} = 17$ h, std $\sigma^{\text{arr}} = 1$ h, average $\tilde{t}^{\text{dep}} = 8$ h, and std $\sigma^{\text{dep}} = 1$ h were assumed based on typical expected residential EV behaviour. Note that each EV connected to the grid is assumed to charge every day between its arrival and departure time.

2) Validation: The model of the original test system, i.e., without EVs or PVs, was validated with real data for the currents in each phase, measured at the head of the feeder, for the period between April 2019 to October 2020, as depicted in Fig. 12. Note that the simulation and actual data are relatively close, with the differences attributable to the fact that actual data shows instantaneous snapshots of the currents flowing when these were measured, whereas the simulation results were produced from hourly power measurements at premises, which are hourly averages of the corresponding instantaneous values.

Charging events for the EVs in the pool were simulated based on the control strategy discussed in Section III-A

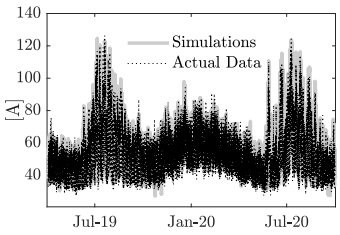


Fig. 12. Head of the feeder base current in phase B; phases R and W show similar behaviour.

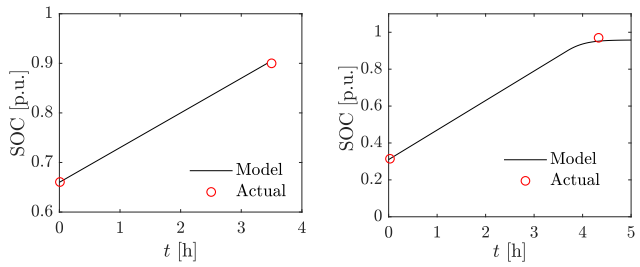


Fig. 13. EV charging events for (a) Tesla Model X and (b) BMW ai3.

and validated against actual data from the Charge the North program [46]. For example, Fig. 13a shows the SOC of a Tesla Model X EV, which starts a charging event with an $SOC_{\rm o} = 0.66$ p.u. reaching an $SOC_{\rm f} = 0.9$ p.u. after 3.5 h, when the charging event ends. Similarly, Fig. 13b shows the SOC of a BMW ai3 EV, which starts a charging event with an $SOC_{\rm o} = 0.31$ p.u. reaching an $SOC_{\rm f} = 0.97$ p.u. after 4.33 h. Note that the model is capable of accurately predicting the final SOC in both cases.

B. Controller training and validation

1) Network Topology: The proposed topologies for the NNs that model Q and μ in the ML-P controller comprise three fully-connected hidden layers with Rectified Linear Unit (ReLU) activation functions at each neuron, and batch normalization layers before each hidden layer [50]. Since Q maps state-action space to Q values, i.e., $Q : \mathcal{S} \times \mathcal{A} \rightarrow Q$, its input layer has as many neurons as the dimension of the state-action space, i.e., $\dim(s_0)+\dim(a_0) = [2(23)+1]+23 = 70$ neurons, which is also the number of neurons per hidden layer of Q. The output layer of Q has one neuron since $\dim(Q) = 1$ with a linear activation function. On the other hand, the input layer of μ has 47 neurons, i.e., $\dim(s_0) = 2(23)+1$, and 21 neurons in the output layer, with tanh activation function. Some relevant training parameters used were: $\gamma_0 = 0.99$, $\tau_0 = 0.001$, size$(\mathcal{R}_0) = 672$ (4 weeks of hourly data), size$(\mathcal{B}_0) = 72$ (3 days of hourly data), $\sigma_{0_o} = 10^{-5}$, $\varepsilon_0 = 1.001$, and learning rates for the critic and actor networks of $Lr_0^c = 2 \times 10^{-5}$ and $Lr_0^a = 1 \times 10^{-5}$, respectively, and were selected based on models developed for similar applications as the one discussed in the paper [14], [30]. The parameters of the reward function in (24) are $\phi_1 = 0.04$, $\phi_2 = 0.001$, $LF_{\rm ref} = 0.75$, and $\xi = 1.2$, and were selected by trial and error to achieve an acceptable LF while minimizing the head of the feeder power. Note that the latter implies that an increment of 20% in the feeder head peak $P_{\max}$ due to EV charging is acceptable, considering that a penetration level of 75% adds 1,542 kW of EV charging capacity, which is significant for a system peak demand without EVs of 750 kW (the maximum capacity of the feeder was not provided).

The topologies of Q_i and μ_i for the secondary controller ML-S$_i$ depend on the the number of EVs served by the transformer i, i.e., the input layer of Q_i and μ_i has $4NE_i+2$ and $3NE_i + 2$ neurons, respectively, and the output layer, 1 and NE_i neurons, respectively, where NE_i is defined in Table I. Each network has two hidden layers with ReLu activation functions at each neuron, and batch-normalization layers before each hidden layer. The output layers of Q_i and μ_i have a linear and tanh activation functions, respectively. The rest of the training parameters used were: $\gamma_i = 0.99$, $\tau_i = 0.01$, size$(\mathcal{R}_i) = 672$ (4 weeks of hourly data), size$(\mathcal{B}_i) = 72$ (3 days of hourly data), $\sigma_{i_o} = 10^{-5}$, $\varepsilon_i = 1.01$, $Lr_i^c = 2 \times 10^{-4}$ and $Lr_i^a = 2 \times 10^{-4}$. The reward function parameters are: $\varphi_1 = 1$, $\varphi_2 = 1$, $\varphi_3 = 0.01$ and $\xi = 1.2$. The parameters of ML-Si controllers were tuned following the same principles as the ML-P controller.

2) Training and Validation: For training purposes, a month (July 2020) of hourly simulations is considered as an episode. The controllers in stages 1 and 2 were trained with 1,000 episodes each, which corresponds to 31,000 days of simulations. At the start of each episode, all EVs' SOC are randomly initialized, thus creating an stochastic environment. Once the model is trained, it is tested in simulations using demand data for August 2022, to assess the controller's ability to act in a different environment. A comparison of the DDPG controller, an uncontrolled Business as Usual (BAU) case, a controller based on conventional Deep Q Learning (DQN) [14], and the base demand without EVs are presented in Fig. 14, where the active power at the head of the feeder is presented. The DQN controller cannot handle multidimensional output and thus, the first-level controller action must be split into NT individual controllers, each one trained based on reward functions combining (24) and (23)-(25). As illustrated in Fig. 14, the DQN controller performed significantly worse than the proposed DDPG controller, the main reason being the lack of coordination between the NT controllers and the overall more complex control framework.

To demonstrate that the intelligent DDPG-based controller has learned charging strategies instead of memorizing charging patterns, the controller is validated by simulations with PV and demand data that was not used in training, i.e., August 2020. Observe that the active power at the head of the feeder for the BAU, and DQN cases exceed the imposed limit of 20% $P_{\max}$ every day, for a total of 171 and 231 hours, respectively. On the other hand, when the DDPG-based controller operates, it exceeds the defined limit in 25 hours only (5 days), successfully maintaining the demand below the limit the rest of the time, thus demonstrating that the ML-P controller is achieving its intended objective. Likewise, the average system LF improves from 0.393 in the BAU case to 0.542 when the controller operates.

In Fig. 15, box plots for the departure SOC of the 114 EVs

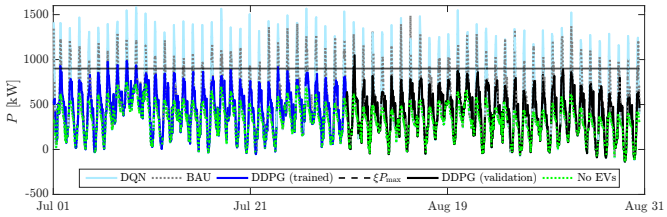


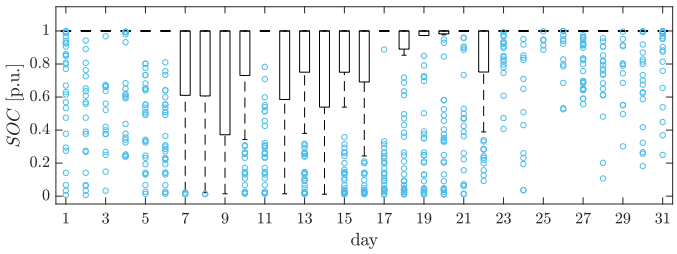
Fig. 14. Head of the feeder active power from July 1 to Aug. 31, 2020.

connected to the feeder are presented for July 2020. Observe that although the BAU case results in a large EV aggregated demand in Fig. 14, the EVs are not always fully charged when they depart, particularly between July 7 to 16. On the other hand, the controller significantly increases the departure SOC on those dates, while keeping reasonable SOC levels on the other days, thus demonstrating that the main objective of the ML-S$_i$ controller is achieved. The distribution transformers in the base system (no EVs) are already overloaded in 193 hours in July 2020, with many other hours being close to this condition. For the BAU case, this number of hours increases to 566, while when the controller operates, the transformers overloaded during 376 hours only, thus demonstrating that the second objective of the ML-S$_i$ is also achieved.

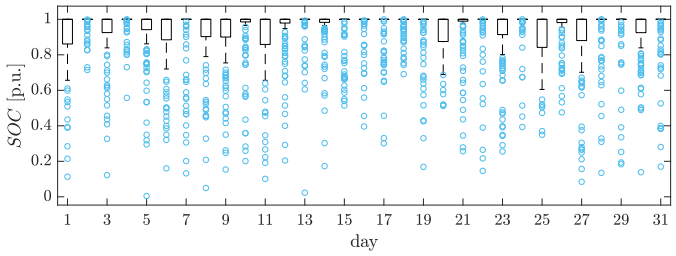
In Fig. 16 box plots for the voltage at the secondary side of all distribution transformers in the feeder for the DDPG controller are presented. Note that the proposed approach is focused on coordinating charging of EVs to mitigate distribution transformers overload, since, as indicated, feeders are typically oversized compared to transformers in practical distribution grids, and considering solar PV injecting only active power by operating as usual at unity power factor. Thus, in the context of practical radial distribution systems, which is the topology of most existing systems as well as the real-world grid used in this study, the problem of distribution transformer overload is synonymous with low voltage profiles. Note that the EV departure and arrival times (5 pm to 8 am with 1-hour std at each time) results in an EV peak demand that does not align with the PV peak production time (around noon), as it is the case in most practical real-world systems. Hence, addressing the transformers overload shifts the EV charging to times closer to PV generation, which should help address overvoltage issues due to PV power injections. In other words, by mitigating the transformer overload, the method implicitly enhances the voltage profile of the system, as shown in Fig. 16. In addition, observe that voltages remain within limits ($v_{\min} = 0.92$ p.u.) for most of the time, even in the BAU case shown in Fig. 10d for the same EV and PV penetration levels, which further reduces the need to consider voltage measurements in the application of the proposed control framework design for the studied test system.

V. Conclusions

This paper presented a practical two-level smart controller for coordinated operation of EVs in a realistic distribution system with PV generation. It was demonstrated that this control, which assumes that the utility sets the charging rates of



(a)



(b)

Fig. 15. EVs' SOC at departure in July 2020 for (a) BAU and (b) control operating.

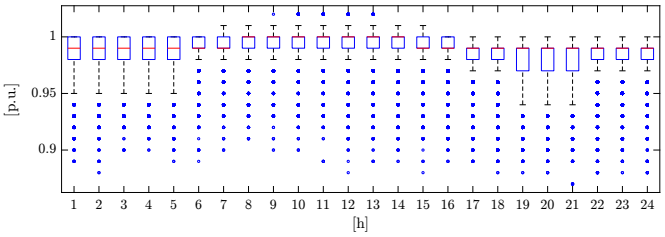


Fig. 16. Transformers' secondary voltages from July 1 to August 31, 2020 for 75% EV and PV penetration under the proposed DDPG control.

the EVs, can effectively learn complex EV charging strategies to achieve aggregated DR objectives, while ensuring fully charged EVs at departure, thus facilitating increased EV penetration in the distribution grid. This justifies the importance of the utility having full control over the EVs, being beneficial to utilities by differing investments, and to customers, who would receive improved service quality, pay lower utility costs, and face less restrictions to connect their EVs to the distribution grid. The control configuration and implementation discussed, can be readily applied to any practical distribution grid. Future work may explore how to incorporate the minimization of individual energy costs of customers when ToU tariffs are applied.

References

[1] "Global market outlook for solar power 2018–2022," Tech. Rep., Solar Power Europe, Brussels, Belgium, 2018. [Online]. Available: https://www.solarpowereurope.org/wp-content/uploads/2018/09/Global-Market-Outlook-2018-2022.pdf

[2] "Ontario's long-term energy plan 2017: delivering fairness and choice," Tech. Rep., Ontario Ministry of Energy, Toronto, Canada, 2017. [Online]. Available: https://files.ontario.ca/books/ltep2017_0.pdf

[3] "Global EV outlook 2019: scaling-up the transition to electric mobility," Tech. Rep., International Energy Agency (IEA), Paris, France, May 2019. [Online]. Available: https://www.iea.org/reports/global-ev-outlook-2019

[4] "Zero-emission vehicles," Government of Canada, Ottawa, Canada, 2020. [Online]. Available: https://tc.canada.ca/en/road-transportation/innovative-technologies/zero-emission-vehicles

[5] R. C. Green II, L. Wang, and M. Alam, "The impact of plug-in hybrid electric vehicles on distribution networks: A review and outlook," *Renewable and Sustainable Energy Reviews*, vol. 15, no. 1, pp. 544–553, 2011.

[6] A. Dubey and S. Santoso, "Electric vehicle charging on residential distribution systems: Impacts and mitigations," *IEEE Access*, vol. 3, pp. 1871–1893, 2015.

[7] A. S. B. Humayd and K. Bhattacharya, "A novel framework for evaluating maximum PEV penetration into distribution systems," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 2741–2751, 2018.

[8] O. Hafez and K. Bhattacharya, "Integrating EV charging stations as smart loads for demand response provisions in distribution systems," *IEEE Trans. Smart Grid*, vol. 9, no. 2, pp. 1096–1106, 2018.

[9] M. Restrepo, C. A. Cañizares, and M. Kazerani, "Three-stage distribution feeder control considering four-quadrant EV chargers," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 3736–3747, 2018.

[10] N. Mehboob, M. Restrepo, C. A. Cañizares, C. Rosenberg, and M. Kazerani, "Smart operation of electric vehicles with four-quadrant chargers considering uncertainties," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 2999–3009, 2019.

[11] S. Alshahrani, M. Khalid, and M. Almuhaini, "Electric vehicles beyond energy storage and modern power networks: Challenges and applications," *IEEE Access*, vol. 7, pp. 99 031–99 064, 2019.

[12] S. Vandael, B. Claessens, D. Ernst, T. Holvoet, and G. Deconinck, "Reinforcement learning of heuristic EV fleet charging in a day-ahead electricity market," *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 1795–1805, 2015.

[13] S. Dimitrov and R. Lguensat, "Reinforcement learning based algorithm for the maximization of EV charging station revenue," in *Int. Conf. Mathematics and Comput. Sci. and Industry*, Varna, Bulgaria, Sep. 2014, pp. 235–239.

[14] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *Proc. Int. Conf. Learn. Representations*, San Juan, Puerto Rico, May 2016, pp. 1–14.

[15] D. Liu, W. Wang, L. Wang, H. Jia, and M. Shi, "Dynamic pricing strategy of electric vehicle aggregators based on DDPG reinforcement learning algorithm," *IEEE Access*, vol. 9, pp. 21 556–21 566, 2021.

[16] D. Qiu, Y. Ye, D. Papadaskalopoulos, and G. Strbac, "A deep reinforcement learning method for pricing electric vehicles with discrete charging levels," *IEEE Trans. Industry Applications*, vol. 56, no. 5, pp. 5901–5912, 2020.

[17] S. Li, W. Hu, D. Cao, T. Dragičević, Q. Huang, Z. Chen, and F. Blaabjerg, "Electric vehicle charging management based on deep reinforcement learning," *J. Modern Power Syst. Clean Energy*, pp. 1–12, 2021.

[18] F. Zhang, Q. Yang, and D. An, "CDDPG: A deep-reinforcement-learning-based approach for electric vehicle charging control," *IEEE Internet of Things J.*, vol. 8, no. 5, pp. 3075–3087, 2021.

[19] Y. Yi, G. Verbič, and A. C. Chapman, "Optimal energy management strategy for smart home with electric vehicle," in *IEEE PowerTech*, Madrid, Spain, June 2021, pp. 1–6.

[20] M. Dorokhova, Y. Martinson, C. Ballif, and N. Wyrsch, "Deep reinforcement learning control of electric vehicle charging in the presence of photovoltaic generation," *Applied Energy*, vol. 301, p. 117504, 2021.

[21] F. Ming, F. Gao, K. Liu, and X. Li, "A constrained DRL-based bi-level coordinated method for large-scale EVs charging," *Applied Energy*, vol. 331, p. 120381, 2023.

[22] B. Zhang, W. Hu, D. Cao, A. M. Ghias, and Z. Chen, "Novel data-driven decentralized coordination model for electric vehicle aggregator and energy hub entities in multi-energy system using an improved multi-agent DRL approach," *Applied Energy*, vol. 339, p. 120902, 2023.

[23] Z. Yang, F. Yang, H. Min, H. Tian, W. Hu, J. Liu, and N. Eghbalian, "Energy management programming to reduce distribution network operating costs in the presence of electric vehicles and renewable energy sources," *Energy*, vol. 263, p. 125695, 2023.

[24] S. Chen, Y. Liu, Z. Guo, H. Luo, Y. Zhou, Y. Qiu, B. Zhou, and T. Zang, "Deep reinforcement learning based research on low-carbon scheduling with distribution network schedulable resources," *IET Generation, Transmission & Distribution*, 2023.

[25] H. Wu, Z. Xu, M. Wang, J. Zhao, and X. Xu, "Two-stage voltage regulation in power distribution system using graph convolutional network-based deep reinforcement learning in real time," *Int. J. Electrical Power & Energy Syst.*, vol. 151, p. 109158, 2023.

[26] F. Kabir, N. Yu, Y. Gao, and W. Wang, "Deep reinforcement learning-based two-timescale volt-var control with degradation-aware smart inverters in power distribution systems," *Applied Energy*, vol. 335, p. 120629, 2023.

[27] G. Chen, H. Zhang, and Y. Song, "Efficient constraint learning for data-driven active distribution network operation," *IEEE Trans. Power Syst.*, 2023.

[28] T. Zhang, L. Yu, D. Yue, C. Dou, X. Xie, and L. Chen, "Coordinated voltage regulation of high renewable-penetrated distribution networks: an evolutionary curriculum-based deep reinforcement learning approach," *Int.J Electrical Power & Energy Syst.*, vol. 149, p. 108995, 2023.

[29] Y. Wang, D. Qiu, Y. Wang, M. Sun, and G. Strbac, "Graph learning-based voltage regulation in distribution networks with multi-microgrids," *IEEE Tran. Power Syst.*, 2023.

[30] X. Sun and J. Qiu, "A customized voltage control strategy for electric vehicles in distribution networks with reinforcement learning method," *IEEE Trans. Ind. Inform.*, vol. 17, no. 10, pp. 6852–6863, 2021.

[31] EPRI. [Online]. Available: https://www.epri.com/pages/sa/opendss

[32] C. Dimas, G. Ramos, L. Caro, and A. C. Luna, "Parallel computing and multicore platform to assess electric vehicle hosting capacity," *IEEE Trans. Ind. Applications*, vol. 56, no. 5, pp. 4709–4717, 2020.

[33] E. Vega-Fuentes and M. Denai, "Enhanced electric vehicle integration in the UK low-voltage networks with distributed phase shifting control," *IEEE Access*, vol. 7, pp. 46 796–46 807, 2019.

[34] N. Knak Neto and L. Piotrowski, "Methodology for analysis of the impact of residential charging of electric vehicles," *IEEE Latin America Trans.*, vol. 17, no. 06, pp. 953–961, 2019.

[35] M. Z. Zeb, K. Imran, A. Khattak, A. K. Janjua, A. Pal, M. Nadeem, J. Zhang, and S. Khan, "Optimal placement of electric vehicle charging stations in the active distribution network," *IEEE Access*, vol. 8, pp. 68 124–68 134, 2020.

[36] K. Montano-Martinez, S. Thakar, S. Ma, Z. Soltani, V. Vittal, M. Khorsand, R. Ayyanar, and C. Rojas, "Detailed primary and secondary distribution system model enhancement using AMI data," *IEEE Open Access J. Power and Energy*, vol. 9, pp. 2–15, 2022.

[37] P. He and A. Khaligh, "Comprehensive analyses and comparison of 1 kW isolated dc–dc converters for bidirectional EV charging systems," *IEEE Trans. Transp. Electrific.*, vol. 3, no. 1, pp. 147–156, 2017.

[38] M. Restrepo, "Smart operation of four-quadrant electric vehicle chargers in distribution grids," Ph.D. dissertation, Elect. Comp. Eng. Dept., University of Waterloo, Waterloo, Canada, 2017.

[39] M. Restrepo, J. Morris, M. Kazerani, and C. A. Cañizares, "Modeling and testing of a bidirectional smart charger for distribution system EV integration," *IEEE Trans. Smart Grid*, vol. 9, no. 1, pp. 152–162, 2018.

[40] O. Tremblay, L.-A. Dessaint, and A.-I. Dekkiche, "A generic battery model for the dynamic simulation of hybrid electric vehicles," in *IEEE Vehicle Power and Propulsion Conf.*, Arlington, TX, Sept. 2007, pp. 284–289.

[41] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[42] P. Radatz, C. Rocha, W. Sunderman, M. Rylander, and J. Peppanen, "OpenDSS PVSystem and InvControl element models," EPRI, Aug. 2020.

[43] C. Rocha, J. Peppanen, P. Radatz, , M. Rylander, and R. Dugan, "OpenDSS: Storage element," EPRI, May. 2020.

[44] "EV everywhere," Bluewave-ai, Ottawa, Canada, 2022. [Online]. Available: https://www.bluwave-ai.com/ev-everywhere

[45] M. Plappert, R. Houthooft, P. Dhariwal, S. Sidor, R. Y. Chen, X. Chen, T. Asfour, P. Abbeel, and M. Andrychowicz, "Parameter space noise for exploration," in *6th Int. Conf. Learning Representations*, Vancouver, Canada, April 2018, pp. 1–17.

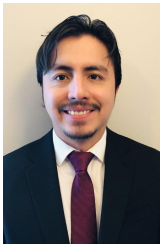
[46] "Charge the north: Results from the world's largest electric vehicle charging study," Tech. Rep. FleetCarma, Kitchener, Canada, 2019.

[47] P. Paudyal, S. Ghosh, S. Veda, D. Tiwari, and J. Desai, "EV hosting capacity analysis on distribution grids," in *IEEE PES GM*, Washington, DC, July 2021, pp. 1–5.

[48] M. A. Awadallah, B. N. Singh, and B. Venkatesh, "Impact of EV charger load on distribution network capacity: A case study in Toronto," *Canadian J. Electrical and Computer Eng.*, vol. 39, no. 4, pp. 268–273, 2016.

[49] "PVWatts® calculator." [Online]. Available: https://pvwatts.nrel.gov/

[50] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Int. Conf. on Machine Learning*, Lille, France, July 2015, pp. 448–456.



Ivan Calero (S'17, M'21) received the diploma in electrical engineering from Escuela Politecnica Nacional (EPN), Quito, Ecuador, in 2008, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2020, where he was Postdoctoral Fellow from 2020 to 2022. From 2009 to 2010, he was with the National Electricity Council of Ecuador. He is currently Project Lead with Power Systems Consultants (PSC), British Columbia, Canada, and a Sessional Lecturer with the University of Waterloo, Ontario, Canada. His research interests include modeling, analysis and control of power systems.



Claudio A. Cañizares (S'85, M'91, SM'00, F'07) is a University Professor, Hydro One Chair, and the WISE Executive Director at the University of Waterloo, where he has been since 1993. His highly-cited research focuses on modeling, simulation, computation, stability, control, and optimization of power and energy systems. He is the IEEE Trans. Smart Gird EIC; Director of the IEEE and PES Boards; a Fellow of the IEEE, the Royal Society of Canada, and the Canadian Academy of Engineering; and has received the 2017 IEEE PES Outstanding Educator Award, the 2016 IEEE Canada Electric Power Medal, and multiple awards and recognitions from PES technical committees.



Mostafa Farrokhabadi (S'12, M'17, SM'22) received his PhD in Electrical and Computer Engineering from the University of Waterloo, ON, Canada, in 2017. He is an Assistant Professor with the Department of Electrical and Software Engineering at the University of Calgary, AB, Canada. His research is broadly focused on smart electrical grids and electric transportation, with a mandate to facilitate the transition to sustainable and universal energy access. He is recognized for his high-impact academic and industrial contributions to electrical distribution systems and microgrids modeling, stability, control, forecasting, and optimization.



Kankar Bhattacharya (M'95, SM'01, F'17) received the Ph.D. degree in electrical engineering from the Indian Institute of Technology, New Delhi, India, in 1993. He was with the Faculty of Indira Gandhi Institute of Development Research, Mumbai, India, from 1993 to 1998, and with the Department of Electric Power Engineering, Chalmers University of Technology, Gothenburg, Sweden, from 1998 to 2002. In 2003, he joined the Electrical and Computer Engineering Department, University of Waterloo, Waterloo, ON, Canada, where he is currently a Professor and the Chair. His current research interests include power system economics and operational aspects. He is a Registered Professional Engineer in the province of Ontario.