Reinforcement Learning-Based Approach for EMT Automation of Large-Scale PV Plants

Qianxue Xia
Energy Science and Technology Directorate
Oak Ridge National Laboratory
Knoxville, USA
xiaq@ornl.gov

Kuldeep Kurte
Computing & Computational Science Directorate
Oak Ridge National Laboratory
Knoxville, USA
kuldeep.kurte@gmail.com

Suman Debnath
Energy Science and Technology Directorate
Oak Ridge National Laboratory
Knoxville, USA
debnaths@ornl.gov

Abstract— **In the pursuit of efficient and precise modeling of large-scale power systems, particularly utility-scale photovoltaic (PV) plants, Electromagnetic Transient (EMT) simulations play a crucial role. As utility-scale PV plants increase in size and complexity, traditional computational methods become inadequate, necessitating more advanced techniques. This paper highlights the progressive efforts made to accelerate EMT simulations. A novel continuous reinforcement learning (RL) strategy is explored to automate the differentiation and categorization of stiff and non-stiff differential algebraic equations (DAEs). The use of stiff and non-stiff integration methods applied to relevant parts of the DAEs assists with the speed-up of the simulations. The paper details the data acquisition, development and offline training of the RL model, leading to its validation that demonstrates a high precision in optimizing simulation methods. The proposed RL promises to significantly enhance the efficacy of EMT simulations, offering a robust framework for the future of power system analysis.**

Index Terms— **Photovoltaic, Electromagnetic transient, Reinforcement learning.**

I. INTRODUCTION

Electromagnetic Transient (EMT) simulations are integral to power systems, particularly as renewables and inverter-based resources are increasingly integrated into the grid. The EMT simulations aid in understanding intricate system dynamics, behaviors, and potential faults. As power systems grow in size and complexity, EMT simulations require more computational power. The escalating demand for accurate and fast simulations has motivated the research community to explore innovative techniques to expedite EMT simulations.

Historically, EMT simulations for large systems like utility scale PV power plants were limited by computational constraints. However, over the years, numerous methodologies have been proposed like using network [1] and node splitting [2] to achieve parallel processing, using hybrid co-simulator comprising EMT and dynamic phasor-based simulators [3], neglecting the switched systems that are present in power electronics, and using high-performance computing (HPC) algorithms to speed up EMT simulations [4][5].

Simulating a vast power plant consisting of numerous PV plants can be time-consuming, in this paper, an advanced numerical simulation algorithms in [6] and [7] are applied to reduce the dimension of matrix inversion and speed up the simulation of up to 326x. The algorithms include numerical stiffness-based segregation, time constant-based segregation, clustering and aggregation on differential algebraic equations (DAEs), and multi-order integration approaches. These algorithms apply multiple discretization algorithms rather than a single discretization algorithm that further reduces the computational burden. However, system configurations may change due to maintenance, grid events, or faults, which can result in changes to the stiff and non-stiff characteristics of the system's states. Consequently, this necessitates adjustments in the discretization algorithms applied to these states. This paper proposes a continuous reinforcement learning (RL) approach for the automatic segregation and dynamic allocation of stiff and non-stiff DAEs, which is crucial for real-time, adaptive simulation management.

Research into the application of RL algorithms within the domain of power systems and power electronics has been gaining attention. The majority of applications address decision-making, control, and optimization challenges in power systems. These include areas such as energy management [8], demand response [9], electricity market [10], operational control [11]. Notably, one publication has explored the use of RL for the control of DC/DC converters [12]. However, to the author's knowledge, there has yet to be literature that specifically applies RL to the EMT simulation of power systems and power electronics.

This paper introduces RL algorithms as a novel approach to the discretization challenges in EMT simulations for large-scale PV plant. It formulates the problem to align with the foundational elements of RL: state, rewards, and actions, tailored to the system's dynamics. A robust offline training setup for RL is designed, and an automated data acquisition process is developed to efficiently manage PV systems with

Research sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U. S. Department of Energy. 2 Energy Science & Technology Science Directorate, Oak Ridge National Laboratory, Knoxville, TN, USA. 1 Computing & Computational Sciences Directorate, Oak Ridge National Laboratory, Oak Ridge, TN, USA. This manuscript has been authored in part by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paidup, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

varying and evolving configurations. The RL model's capability to handle the diverse characteristics and operational changes of these systems without manual intervention marks a significant advancement in EMT simulation and is being performed for the first time. This contribution not only streamlines the simulation process but also enhances the adaptability and scalability of EMT simulations in response to the increasing complexity of large-scale IBRs connected grid system.

II. EMT Simulation for Large-Scale Power Plant

The simulation of large-scale power plants, which can include several PV systems, is inherently time-consuming. The challenge is compounded when frequent changes in PV configuration occur due to maintenance, grid events, or faults. This paper discusses the application of simulation algorithms to streamline this process.

A. PV System Architecture

A large-scale PV plant may consist of different types of PV systems. The PV system considered in this paper consists of a dc-dc boost converter that interfaces the PV panels and the two-level three phase ac-dc inverter, the ac side of the inverter is connected with the ac grid through a LCL filter.

The dual control of the boost converter includes an outer loop input voltage control and an inner loop control of the input side inductor current. The ac-dc inverter controls the dc-link voltage and reactive power sent to the ac grid and generates the dq current references for the inner dq current control. The PV system is shown in Fig. 1.

B. EMT Models

The dynamic equations representing the k^{th} PV system in a large PV plant are expressed as follows:

$$C_{pv}\frac{dv_{cpv,k}}{dt} = -\frac{1}{R_{pv}}v_{cpv,k} + i_{pv,k} - i_{L,pv,k} \tag{1}$$

$$\begin{aligned} L_{pv}\frac{di_{L,pv,k}}{dt} = &-R_{L,pv}i_{L,pv,k} + v_{cpv,k} \\ &- v_{\mathrm{dc},k}\{S_{2,dc,k}(1 - \mathrm{S}_{1,dc,k}) \\ &+ (1 - S_{2,dc,k})(1 - S_{1,dc,k})\mathrm{sgn}(i_{L,pv,k})\} \end{aligned} \tag{2}$$

$$\begin{aligned} C_{total}\frac{dv_{dc,k}}{dt} = &-\frac{v_{dc,k}}{R_{dc}} - \frac{v_{dc,k}}{R_{out,pv}} \\ &+ i_{L,pv,k}\{S_{2,\mathrm{dc,k}}(1 - S_{1,\mathrm{dc,k}}) \\ &+ (1 - S_{2,\mathrm{dc,k}})(1 - S_{1,\mathrm{dc,k}})\mathrm{sgn}(i_{L,pv,k})\} \\ &- i_{j,ac,k}\{S_{1,\mathrm{j,ac,k}}(1 - S_{2,\mathrm{j,ac,k}}) \\ &+ (1 - S_{2,\mathrm{j,ac,k}})(1 - S_{1,\mathrm{j,ac,k}})\mathrm{sgn}(i_{j,ac,k})\} \end{aligned} \tag{3}$$

$$\begin{aligned} L_{1,ac}\frac{di_{j,ac,k}}{dt} = &-R_{1,ac}i_{j,ac,k} \\ &+ \frac{v_{\mathrm{dc},k}}{2}\{S_{1,\mathrm{j,ac,k}}(1 - S_{2,\mathrm{j,ac,k}}) \\ &- S_{2,\mathrm{j,ac,k}}(1 - S_{1,\mathrm{j,ac,k}}) \\ &- (1 - S_{2,\mathrm{j,ac,k}})(1 \\ &- S_{1,\mathrm{j,ac,k}})(2\mathrm{sgn}(i_{j,ac,k}) - 1)\} - v_{j,ac,fil,k} \end{aligned} \tag{4}$$

$$C_{1ac}\frac{dv_{j,ac,fil,k}}{dt} = -\frac{1}{R}v_{j,ac,fil,k} + i_{j,ac,k} - i_{j,ac,fil,k} \tag{5}$$

$$L_{2,ac}\frac{di_{j,ac,fil,k}}{dt} = -R_{2,ac}i_{j,ac,fil,k} + v_{j,ac,fil,k} - v_{\mathrm{j,grid}} \tag{6}$$

where $\mathrm{sgn}(x) =$

$$\begin{cases} 0 & if\ x < 0 \\ 1 & if\ x > 0 \end{cases} \quad \forall\, y \in (p,n), \forall\, j \in (a,b,c), \forall\, k \in [1,N]$$

$$C_{total} = C_{out\_pv} + \frac{C_{dc}}{2}$$

C. Numerical Stiffness and Time Constant

The "sgn" function introduces stiffness due to its discontinuous nature. This discontinuity leads to rapid changes in the values of the solutions, meaning that the equations have components that are varying at different rates. Solvers often have trouble at these points, and it may require very small timesteps to accurately capture the behavior.

In order to address stiffness caused by the "sgn" function, an event handling method is introduced in [13] with its hybrid discretization algorithms. And the nonlinearity of the sgn function is handled by the hysteresis relaxation technique.

The DAEs representing the dynamics of the system are categorized into stiff DAE and nonstiff DAE based on the time constant , following protocols from reference [6], especially for equation (1) and (3), (5). For stiff DAEs, a stiff-decay discretization algorithm, such as backward Euler, is preferred, whereas non-stiff DAEs are treated with forward Euler for discretization.

The circuit configuration of the PV system is not always static and unchanged. It may undergo changes in circuit parameters like input capacitance and inductance due to fault, capacitor degradation, capacitor replacement, and may feature different configurations like different power rating, voltage rating, different PV panels. It is also likely that they may experience circuit upgrade like changing from an ac-dc inverter to a dc-dc boost converter followed with ac-dc inverter, or a dual active bridge and ac-dc inverter or changing the filter from LC to LCL. In one PV plant, there are hundreds of PV systems and if one were to perform segregation of stiff and non-stiff DAE manually, it would be extremely time consuming, especially if it needs to be done online and requires limited time. Thus, continuous RL methods are proposed to automate mapping and partitioning in real-time of the models in dynamic simulations to parallelizable resources.

Given the multitude of systems within a single plant, manual segregation of stiff and non-stiff DAEs is impractical, especially when real-time adjustments are needed. Continuous RL is proposed to automate the mapping and partitioning of models in dynamic simulations to parallelizable resources, thus optimizing the process.

III. Reinforcement Learning

A. Problem Definition

The objective is to enhance the speed of numerical simulations—specifically DAEs—by automating the decomposition or disaggregation process based on the characteristics of the DAEs. This automation aims to scale the process for large-scale problems without the need for manual selection of DAE properties. The challenge involves continuous, sequential decision-making, where a RL agent interacts with the simulation environment by:

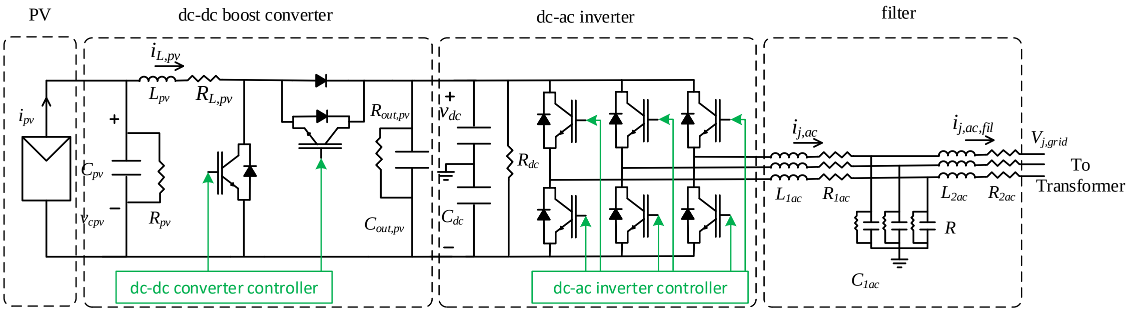


Fig. 1. PV system.

- Observing the current state.
- Taking actions that influence the simulation, such as those depicted in Fig. 2.
- Transitioning the environment to a new state based on these actions.
- Receiving scalar feedback (reward) from the environment.
- Refining its policy to improve performance.

The RL agent's goal is to discover an optimal policy that maximizes long-term rewards. In this context, the state, the environment, and the reward structure are central to the RL process, and they are specifically defined to address the issue of manually selecting DAE discretization parameters and simulation time steps.

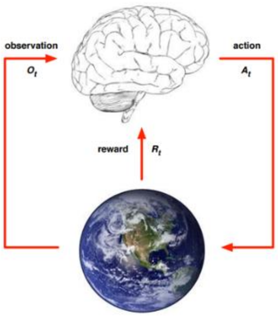


Fig. 2. Reinforcement learning.

1) State

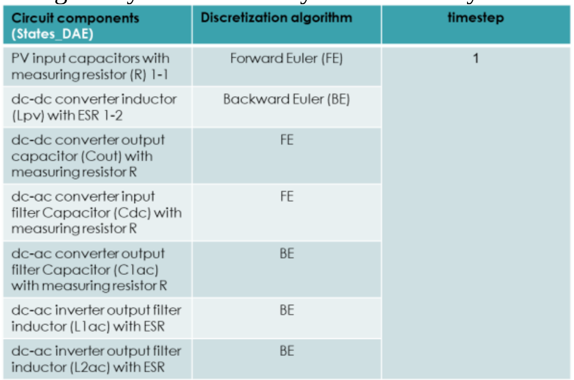
The states represent the conditions and properties of the circuit that influence its dynamic behavior and are relevant to the learning process:

- *Circuit components (States_DAE):* the components responsible for the dynamic change in the circuit configuration
- *Stiffness*: stiffness is associated with the particular terms in the system of DAEs that cause rapid variation in the solution of DAEs. At times, improper selection of numerical methods to solve such terms might jeopardize the numerical stability of the system.
- *Time constants:* one of the time-domain parameters to evaluate the system performance using the circuit parameter values.
- *Minimum time step requirement:* the minimum time step in the simulation needed to accurately capture the dynamics of the states.

2) Action

The action space comprises two primary decisions that the agent can make:

- *DAE Discretization*: This involves choosing a discretization method for the DAEs, essentially converting continuous-time models into discrete-time counterparts for simulation.
- *Simulation Time Step:* This is the selection of time increments for updating the simulation's state, critical for capturing the dynamics of the system accurately.



Circuit components (States_DAE)	Discretization algorithm	timestep
PV input capacitors with measuring resistor (R) 1-1	Forward Euler (FE)	1
dc-dc converter inductor (Lpv) with ESR 1-2	Backward Euler (BE)	
dc-dc converter output capacitor (Cout) with measuring resistor R	FE	
dc-ac converter input filter Capacitor (Cdc) with measuring resistor R	FE	
dc-ac converter output filter Capacitor (C1ac) with measuring resistor R	BE	
dc-ac inverter output filter inductor (L1ac) with ESR	BE	
dc-ac inverter output filter inductor (L2ac) with ESR	BE	

Fig. 3. One type of action of the problem.

For the PV system illustrated in Fig. 1, we observe seven distinct states. Actions related to the DAE discretization are represented as binary variables, with the simulation time steps considered being 1, 4, and 10 microseconds. By applying one-hot encoding to represent each possible action uniquely, we have a total action space size of 383 distinct possibilities.

3) Reward

The reward function is continuous and designed to incentivize the following outcomes:

- *System Stability:* Ensuring the system remains stable following the chosen actions.
- *Harmonic Level:* Minimizing the level of harmonic distortion in the system's output.
- *Time Efficiency:* Reducing the simulation's runtime without compromising the accuracy and stability of the results.

The continuous nature of the reward facilitates detailed discrimination among the outcomes of various actions, directing the agent toward the most advantageous behavior as it learns.

B. Automation via RL

RL algorithms come in various forms. Initially, the simple Multi-Armed Bandit algorithm is considered, which focuses solely on actions and rewards without interacting with the environment. However, this is deemed unsuitable for our purposes, as the environment—our circuit topology—provides varying states that must be considered. We then explore Contextual Bandits, which extend the Multi-Armed Bandit problem by incorporating context, or state information, into the decision-making process. Here, an agent selects from a set of possible actions based on the current context and receives a reward dependent on both the action taken and the context presented. The agent's task is to learn the optimal action for each context to maximize cumulative rewards, despite initially unknown reward distributions. This necessitates a balance between exploration (testing various actions) and exploitation (selecting the best-known action for a given context).

Fig. 4 illustrates an RL system in which the agent interacts with a simulated environment. The 'DAE properties' constitute the state or context for the agent. With this context, the agent chooses an action that impacts the system's dynamics. The consequent feedback—reward—is based on the "convergence of simulation states," reflecting the suitability of the chosen action against the expected outcomes. The agent aims to learn actions that maximize its overall rewards, using the insights from the DAE properties.

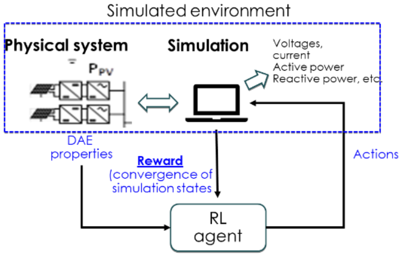


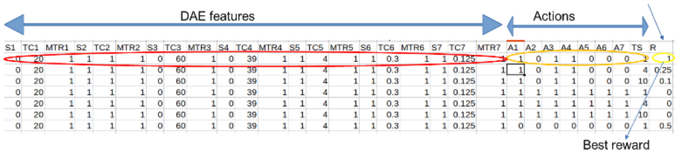
Fig. 4. Contextual Bandits application in PV system simulation.

The ϵ-greedy algorithm with a decaying exploration rate is selected for its balance between exploring new actions and exploiting known good actions. As the agent gains more experience, the exploration rate decays, leading the agent to increasingly rely on the best-known strategies. This approach helps ensure that the agent remains sufficiently exploratory in the early stages, while gradually becoming more exploitative as it converges towards an optimal policy.

IV. Reinforcement Learning Based EMT Automation

A. Data Acquisition

The creation of a dataset is a critical step for training an RL model. The dataset must reflect a variety of operational conditions and parameters that a PV system might encounter. For this purpose, different configurations of a PV system are used, varying the power and voltage ratings as well as the type of PV panel. Changes in the type of PV panel will also alter the associated model parameters. The configuration of the number of modules in series and parallel is dependent on the system's ratings, which influences the circuit parameters such as inductance and capacitance, as well as the controller gains. An automated process is designed to configure the PV parameters, circuit parameters, and some controller parameters to facilitate data acquisition. The dataset structure is shown in the Fig. 5 below. It contains the DAE properties of all state variables, actions for a set of DAE properties, and the rewards for those actions.


DAE features
Actions
Best reward

S1	TC1	MTR1	S2	TC2	MTR2	S3	TC3	MTR3	S4	TC4	MTR4	S5	TC5	MTR5	S6	TC6	MTR6	S7	TC7	MTR7	A1	A2	A3	A4	A5	A6	A7	TS	R
0	20	1	1	1	1	0	60	1	0	39	1	1	4	1	1	0.3	1	1	0.125	1	1	0	1	1	0	0	0	1	1
0	20	1	1	1	1	0	60	1	0	39	1	1	4	1	1	0.3	1	1	0.125	1	1	0	1	1	0	0	0	4	0.25
0	20	1	1	1	1	0	60	1	0	39	1	1	4	1	1	0.3	1	1	0.125	1	1	0	1	1	0	0	0	10	0.1
0	20	1	1	1	1	0	60	1	0	39	1	1	4	1	1	0.3	1	1	0.125	1	1	1	1	1	1	1	1	1	0
0	20	1	1	1	1	0	60	1	0	39	1	1	4	1	1	0.3	1	1	0.125	1	1	1	1	1	1	1	1	4	0
0	20	1	1	1	1	0	60	1	0	39	1	1	4	1	1	0.3	1	1	0.125	1	1	1	1	1	1	1	1	10	0
0	20	1	1	1	1	0	60	1	0	39	1	1	4	1	1	0.3	1	1	0.125	1	0	0	0	0	0	0	0	1	0.5

Fig. 5. Dataset structure.

B. Offline Training

The following is the pseudocode for the Decay epsilon greedy based contextual bandit algorithm. Assuming there are k number of actions, this algorithm will keep a list of k regressors (oracles).

```
Algorithm: Decay Epsilon-Greedy
    Inputs exploration probability $p \in (0,1)$, decay rate $d \in (0,1)$,
    oracles $\hat{f}_{1:k}$, training_frequency
      Initialize history H for each oracle to empty
      Initialize exploration probability p
    For each successive round t with context $x^t$ do
        With probability (1-p) and t > training_frequency:
            Select action $a^t = argmax_k \hat{f}_k(x^t)$ using the oracles
        Otherwise:
            Select action $a^t$ uniformly at random from 1 to k
        Execute action $a^t$ and observe reward $r^t$
        Update history $H$ by adding the new observation $(x^t, r^t)$
        Update oracle $\hat{f}_{a^t}$ with the new history H
        Update exploration probability $p = p \times d$
    End for
```

Fig. 6 depicts the contextual bandit setup, where the agent interacts with an environment that presents contexts and dispenses rewards based on the agent's actions. Rewards are probabilistic, drawn from an unknown distribution. The agent's goal is to learn a policy that maximizes cumulative long-term rewards, navigating the balance between exploring new actions and exploiting known profitable ones.

For training purposes, instead of a live environment, tabulated data are used to emulate the conditions the agent will encounter. This data-driven approach allows us to simulate interactions and train the agent effectively.

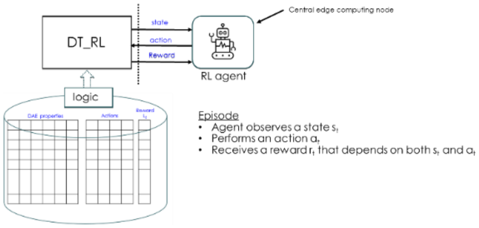

Central edge computing node
DT_RL
state
action
Reward
RL agent
logic
DAE properties
Actions
Reward r_t
Episode
• Agent observes a state s_t
• Performs an action a_t
• Receives a reward r_t that depends on both s_t and a_t

Fig. 6. RL offline training set up.

C. Model Evaluation and Deployment

This study focuses on the offline deployment of the RL model. The exploration of its online deployment will be part of the future work.

Fig. 7 illustrates the decay of the exploration probability epsilon over time steps. It shows that the policy starts with a higher likelihood of exploration, which decreases over time as the agent presumably gains more knowledge about which actions are better in different states, thus shifting the balance from exploration to exploitation.

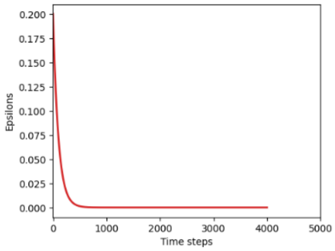


Fig. 7. Decay of Exploration Probability.

Subsequent to the offline training phase, the model's capabilities were evaluated by the cumulated mean reward and its ability to predict actions for a range of input states. Fig. 8 displays the progression of cumulative mean reward over time for an epsilon-greedy strategy in a RL context. The initial high slope suggests a rapid gain in reward, which levels off as the agent likely starts to exploit more than explore, reflecting a stabilization in learning.

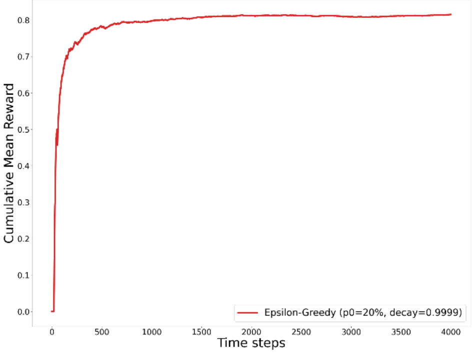


Fig. 8. Cumulative mean reward.

The model demonstrates its ability by determining the optimal actions for various states. Take, for example, a state delineated by the array [1, 2, 1, 1, 0.215, 1, 0, 60, 1, 0, 12, 1, 1, 3, 1, 1, 0.41, 1, 1, 0.016, 1], which maps to a series of state variables denoted as [S1, TC1, MTR1, S2, TC2, MTR2, S3, TC3, MTR3, STR3, ... S7, TC7, MTR7]. Here, 'S' represents the stiffness values, 'TC' stands for the time constants, and 'MTR' indicates the minimum time requirements for the state variables. Each element in the sequence provides a snapshot of the system's state, detailing the stiffness, time constant, and minimum time requirement for each variable, which the model uses to compute the most effective action sequence.

In this specific case, the model predicts the optimal action to be [0, 0, 1, 1, 0, 0, 0, 1], after being encoded, translates to the selection of numerical methods for various components in a PV system. The first 7 variables are for the discretization selection,"1"s in the array indicate the application of a forward Euler method for the dc-dc converter output capacitor, and the dc-ac converter input filter capacitor. Conversely, a backward Euler method is selected for the remaining state variables and is shown as “0”. The last “1” in the array denotes that the timestep for these actions is set to 1 us, which aligns with the configuration depicted in Fig. 3. The RL successfully chooses the optimal action for given contexts.

V. Conclusion

The paper proposed a promising application of RL to streamline EMT simulations for hundreds of PV system in a large PV plant. By employing continuous RL for the automatic differentiation and real-time partitioning of stiff and non-stiff DAEs, a significant reduction in computational load and time taken to simulate can be achieved. The offline training and subsequent evaluation of the RL model demonstrated high accuracy in selecting optimal discretization methods and simulation timestep for various system configurations. The groundwork laid by this research suggests a transformative potential for RL in the domain of power systems, where the necessity for real-time, adaptive simulation management is becoming increasingly crucial. As the paper indicates, further exploration into the online deployment and training of these RL models is poised to enhance the efficiency and reinforce the reliability and robustness of EMT simulation.

References

[1] M. Armstrong, J. R. Marti, L. R. Linares, and P. Kundur, “Multilevel mate for efficient simultaneous solution of control systems and nonlinearities in the OVNI simulator,” *IEEE Transactions on Power Systems*, vol. 21, no. 3, pp. 1250–1259, 2006.

[2] C. Yue, X. Zhou, and R. Li, “Node-splitting approach used for network partition and parallel processing in electromagnetic transient simulation,” vol. 1, 12 2004, pp. 425 – 430 Vol.1.

[3] K. Mudunkotuwa, S. Filizadeh, "Co-simulation of electrical networks by interfacing EMT and dynamic-phasor simulators", *Electric Power Systems Research*, Volume 163, Part A, 2018.

[4] S. Subedi et al., “Review of methods to accelerate electromagnetic transient simulation of power systems,” *IEEE Access*, vol. 9, pp. 89714–89731, 2021.

[5] F. Li et al., “Review of real-time simulation of power electronics,” *J. Modern Power Syst. Clean Energy*, vol. 8, no. 4, pp. 796–808, Jul. 2020.

[6] S. Debnath and J. Choi, "Electromagnetic Transient (EMT) Simulation Algorithms for Evaluation of Large-Scale Extreme Fast Charging Systems (T& D Models)," in *IEEE Transactions on Power Systems*, vol. 38, no. 5, pp. 4069-4079, Sept. 2023.

[7] J. Choi and S. Debnath, "Electromagnetic Transient (EMT) Simulation Algorithm for Evaluation of Photovoltaic (PV) Generation Systems," 2021 IEEE Kansas Power and Energy Conference (KPEC), Manhattan, KS, USA, 2021, pp. 1-6.

[8] Z. Q. Wan, H. P. Li, H. B. He, and D. Prokhorov, “Model-free real-time EV charging scheduling based on deep reinforcement learning, ” *IEEE Transactionson Smart Grid*, vol. 10, no. 5, pp. 5246–5257, Sep. 2019.

[9] R. Z. Lu and S. H. Hong, “Incentive-based demand response for smart grid with reinforcement learning and deep neural network, ” *Applied Energy*, vol. 236, pp. 937–949, Feb. 2019.

[10] T. Chen and W. C. Su, “Indirect customer-to-customer energy trading with reinforcement learning, ” *IEEE Transactionson Smart Grid*, vol. 10, no. 4, pp. 4338–4348, Jul. 2018.

[11] L. Xi, J. F. Chen, Y. H. Huang, T. L. Xue, T. Zhang, Y. N. Zhang, “Smart generation control based on deep reinforcement learning with the ability of action self-optimization, ” *Scientia Sinica Information is*, vol. 48, no. 10, pp. 1430–1449, Oct. 2018.

[12] Soonhyung Kwon, Changwoo Yoon, Young-Il Lee, "Practical Implementation Method of Reinforcement Learning for Power Converter", *IFAC-PapersOnLine*, Volume 55, Issue 9, 2022

[13] S. Debnath and M. Chinthavali, “Numerical-stiffness-based simulation of mixed transmission systems,” *IEEE Trans. Ind. Electron*., vol. 65, no. 12, pp. 9215–9224, Dec. 2018.