iEnergy

ARTICLE

https://doi.org/10.23919/IEN.2022.0009

Towards an AI-friendly cross-timescale simulation and analysis platform for electric distribution systems

Dongqi Wu, Rayan El Helou and Le Xie

ABSTRACT

Substantial changes are occurring in electric distribution systems due to ambitious targets towards carbon-neutrality in many regions around the world. One of the key challenges is how to analyze the interactions of massive amount of energy end-users with the electric distribution grid operator. In this paper, we introduce a comprehensive simulation platform, AI4Dist, that is capable to perform a wide collection of distribution system studies that capture multiple timescales ranging from market planning to transient event analysis. AI4Dist is designed to effortlessly integrate with off-the-shelf machine learning packages and algorithm implementations. We envision that AI4Dist will serve as a platform to empower researchers with different expertise to contribute to the development of low carbon electricity sector.

KEYWORDS

Power distribution system, machine learning, simulation platform.

The ambitious goal of decarbonizing the electric power sector has inspired and accelerated the adoption of many new technologies in power distribution systems. Carbon-free energy sources such as wind and solar photo-voltaic (PV) are quickly gaining more momentum in the energy mix composition. The share of electric vehicles (EVs) in the consumer market has increased significantly over the past few years due to their lower millage cost, better driving experience and low environmental impact. Demand side management programs such as price-sensitive load and incentivized demand response are emerging across the globe to compensate uncertainties and difficulties introduced by the new changes to conventional energy paradigm that have been used for decades.

Along with the rapid development of these grid-edge components, the uncertainty they bring began to emerge as a problem for operators in power grid companies. For example, the output of distributed renewable generators can vary greatly in a few minutes due to weather change and greatly disrupt the power flow pattern; the charging capacity of EVs[1] can reach 3 times the total power consumption from everything else in homes while their schedules are tied to the user-specific life and mobility patterns which are not known to distribution grid operators. In contrast, the second-largest component of household power consumption, the heating, ventilation and air conditioning (HVAC) systems, are closely related to climate and thus much easier to predict compared to EVs. Demand response programs that encourage users to adjust their consumption time to adapt changes in the grid can help to alleviate the load-side uncertainty, but the inherent human-factor and difference in price sensitivity among customers are still difficult to rely upon. These problems are particularly obvious in power distribution systems compared to the transmission grid, which has a much higher inertia and size to damp out the uncertainties.

Recent developments in data science and artificial intelligence (AI) have shown enormous potential of offering solutions to problems that accompanies the transition and evolution to low-carbon energy sector. AI-based approaches have demonstrated astonishing performance in many aspects of power distribution system and micro-grid operation, examples including renewable/load[2,3] and market[4] forecasting, voltage regulation[5], monitoring[6], protection[7] and demand response[8]. Data-driven algorithms are also seen in many applications such as load modelling[9] and fault detection[8].

Despite the increasing popularity of developing AI and data science based solution on power system problems, there still exists a major roadblock in the path of converting these technologies into real-world applications: the algorithms in literature need to be evaluated and compared on a common simulation benchmark. However, existing power distribution system simulators are not ready for this task as they lack the key features needed to accurately re-produce the dynamics of future distribution systems. First, they are not designed to integrate with commonly-used AI and data science software packages, which requires every researcher to develop a work-around implementation that is mostly exclusive to their own algorithm and make cross-comparison and benchmarking impossible. Moreover, to simulate systems with multiple AI-based components serving different roles and have dynamics with various timescales ranging from micro-seconds to minutes, the simulator will need to account for the interaction among decision-makers as their behavior depend on not only exogenous static inputs, as in the case of conventional simulation, but also the policy and action of other decision makers in the same circuit.

There already exist several recognized tools for simulation power distribution systems. Popular ones include OpenDSS[10] and HELICS[11], both are free, open-source and can be controlled through Python. Based upon these original simulation engines, various customized expansions such as MA-OpenDSS[12] and OpenDSS-Wrapper[13] are also developed to better suit the need to

Department of Electrical and Computer Engineering, Texas A&M University, College Station, Texas, USA
Address correspondence to Le Xie, le.xie@tamu.edu

study specific problems, high DER penetration and control in these cases. AMES[14] is an open-source simulator developed in Java that focuses on wholesale power market analysis. Grid2OP[15] is a pioneer among the efforts to introducing reinforcement learning (RL) into power system by providing a well-defined formulation for real-time transmission grid operation. Our previous work OpenGridGym[16] provides a Python-based toolkit to test possible designs of market mechanisms in distribution systems and strategies of market participants. We propose a similar, more general simulation platform for distribution grids.

Suggested contributions

We aim to provide a simulation and analysis platform, AI4Dist [17], that fills the gap between conventional power distribution simulators and the needs to model, study and evaluate the performance of potential AI and data science based grid components. We propose a generalized formulation to model inter-dependent decision makers controlling and monitoring various components in power distribution networks. In contrast to most conventional simulators that assumes a deterministic behavior from grid components (dispatch, loads, relays, reactive power support devices, etc.), AI4Dist models them as *Agents* that can be programmed using any AI or data science based methods and seamlessly fit into the simulation workflow. Under this multi-agent framework, the effect of all types of decision makers on the physical grid and on others' actions can be easily captured. Another unique feature of AI4Dist is its ability to run integrated scenarios that involves user-modelled decision makers that act at different timescales ranging from market intervals (10-min scale) to fault study (millisecond) levels. This allows users to discover induced events between decision makers with different timescales that are usually not considered during conventional studies. For instance, a surge of load capacity due to EV charging could cause a sudden system undervoltage; or a persisting fault that caused the tripping of part of the circuit could have a significant impact on the market prices. AI4Dist provides a consistent and easy-to-use platform for the development, evaluation and benchmarking of control algorithms for various distribution grid components.

1 Design of AI4Dist

1.1 Design Objectives

As power distribution systems have started the transition to incorporate distributed energy resources (DERs), electric vehicles and demand side management technologies at unprecedented speed, various data-driven and artificial-intelligence based solutions are proposed to tackle the challenges that arise along with this transition. A suitable simulation test-bed that can easily integrate with these new solutions and provide accurate simulation results is necessary to advance the integration of data-driven solutions toward real-world applications. We envision AI4Dist to be a comprehensive platform that includes the physical models of many new elements in modern distribution systems and provide a plug-and-play style interface for external control algorithms. Figure 1 shows the concept of AI4Dist.

AI4Dist is designed to have the following features that we consider to be indispensable to perform realistic simulation for modern power distribution systems:

User friendliness for data science and AI expertise

The difficulties to develop, evaluate and benchmark data science or AI methods using conventional distribution system simulators mainly lie in three aspects: (1) The automation capability of most simulators are not efficient enough as users need to design and implement workflow control scripts that instruct the simulators to run repetitive simulations. Most studies using conventional analytical approaches usually consider only a few representative scenarios. However, the amount of data required to train and calibrate data-driven methods is enormous and must be generated through simulating the system under different operating conditions thousands of times; (2) The implementation of data science and AI algorithms relies heavily on existing libraries such as stable-baselines[18] which have dependencies and requirement such as machine learning back-ends (e.g. Tensorflow, PyTorch) that are not directly compatible with power system simulators; (3) Most conventional simulators offer very limited expandability for controlling circuit elements using external algorithms. The control of most circuit elements (e.g. loads, relays, voltage regulators) is determined by built-in models, getting around them typically require tedious work that is confined to the specific experiment. AI4Dist allows users to efficiently create, modify and duplicate agents that can be integrated into the simulation workflow automatically.

Interaction among interdependent agents

In the analysis and simulation of conventional power distribution systems, most components are assumed to be completely passive

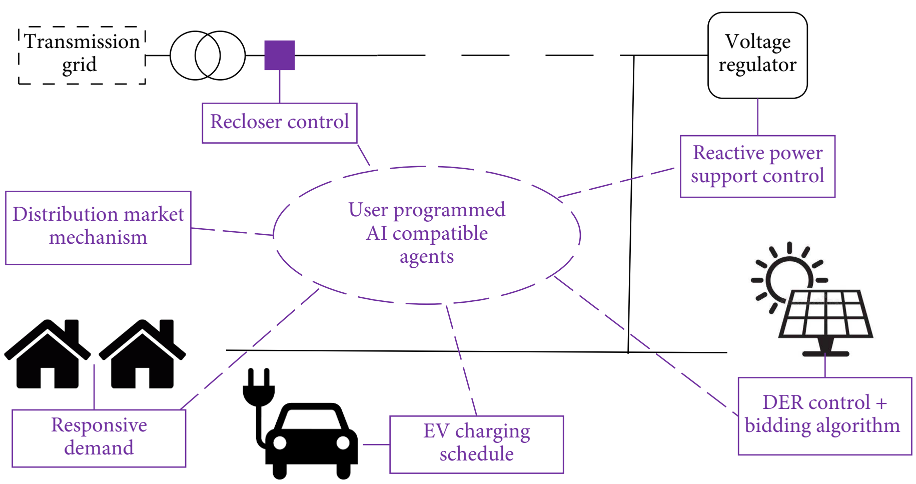


Fig. 1 Envisioned concept of AI4Dist.

(e.g. load, capacitors) or have deterministic closed-from models (e.g. voltage regulators, relays). This assumption becomes less and less appropriate as distribution systems become more dynamic and decentralized. For example, loads are usually modelled as fixed power/current/impedance, or a combination of these three components. These parameters are not related to conditions in the distribution circuit. With the introduction of demand-side management technologies, load capacities might change in real-time depending on their prices, which is related to the other DERs in the circuit, or incentives, which can be tied to supply and demand balance in the bulk transmission system. The responsive component in load capacity becomes even more significant with the increasing popularity of EV, which represents both very high peak power and a large part of total household energy consumption. Accurate modeling of the active interaction between most system components and other parts of the distribution circuit is crucial while being a missing feature in most existing simulators.

Multi-timescale simulation

A necessary prerequisite for modelling the interaction among different types of active decision makers is the coordination between different timescales, as the behaviors of many active decision makers are affected by grid data computed with different time resolution. For example, market clearings usually occur periodically with a relatively long interval (15 min in many transmission systems), while the behaviors of price sensitive loads (EVs) and generators (gas/diesel turbines) depend heavily on prices. Hence, to perform simulation that contains accurate load dynamics, one must also generate the price data by running market simulation that also depends on inputs from generators. Figure 2 shows an example of DER output in two timescales, in which the real-time physical fluctuation around the settled capacity within market intervals is significant and must be accounted for separately. Such scenarios require the simulator to be able to run several different timescales in parallel to correctly capture all the dynamics in the distribution system.

1.2 Program design and structure

We present the program design of our simulation platform that

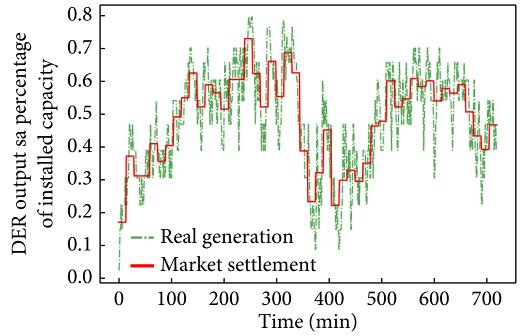


Fig. 2 Fluctuation of DER output under different timescales.

meets the various requirements to simulate modern power distribution systems summarized in the previous subsection. A block diagram showing the architecture is presented in Figure 3, which will be explained in detail below.

Architecture

AI4Dist takes a modular approach in its workflow, i.e. the input data, user-defined decision maker models and simulation engines can all run individually or simultaneously in any combinations. Users may implement their algorithms by providing an *Agent* that controls an element in the distribution circuit. Each agent communicates with the simulation engine, the *environment* , by accessing and storing to certain pre-defined *fields* in the agent structure during the simulation. The environment passes the states of the distribution circuit that each Agent need to measure, also by storing into the fields that correspond to observations in each Agent object. Agents then compute their *actions* , i.e. change of parameters in the circuit, based on the observations they receive. Take a conceptual voltage regulator as example, its duty is to regulate the terminal voltage at which it is connected to the distribution system by adjusting its reactive power output. In this case, the observations would be the terminal voltage and the actions will be the net reactive power output. This framework provides users with maximum freedom, as they can implement a mapping between observations and actions in a completely standalone program, which may use any libraries or even external software as

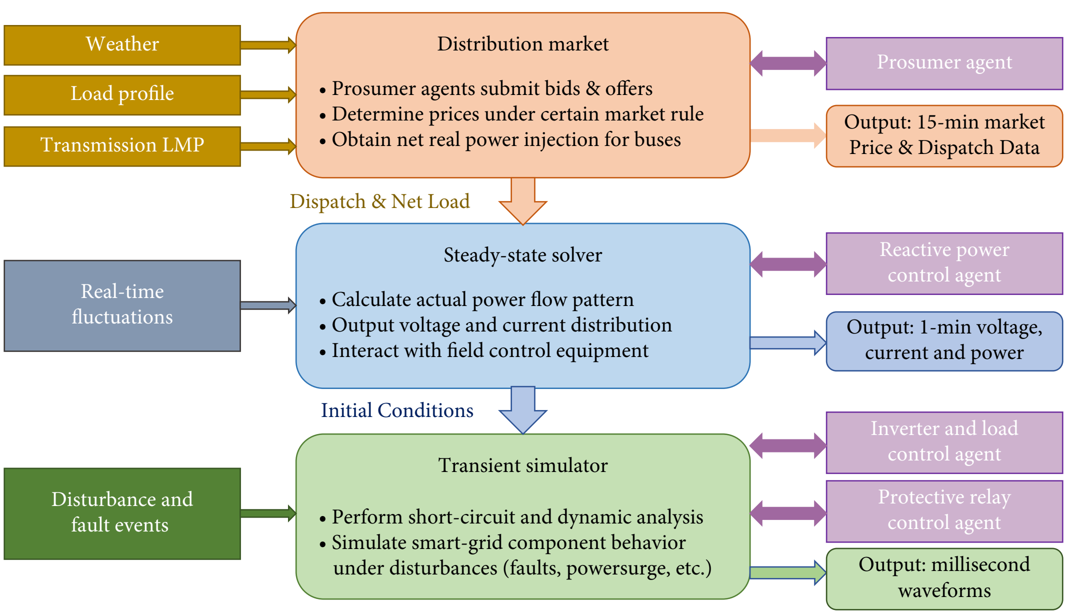


Fig. 3 Architecture of AI4Dist.

long as the final actions are given in the end.

Environment and computational engine

We choose Python as the codebase of AI4Dist as it has several unique advantages among all alternatives. First, Python is widely considered as the most common and convenient language among the machine learning community. Many learning, optimization and computing toolboxes are readily available for use. Second, several distribution simulators already have a Python interface that can be accessed to control the simulation workflow. We choose for AI4Dist the use of the open-source software OpenDSS[10] as the computational engine mainly because it performs three-phase unbalanced power flow and short-circuit calculations, and because the authors believe it is easier to interact with via Python than existing alternatives. On top of the physical grid simulation from OpenDSS, we built a market layer that can be interfaced with user-defined market participants to determine generator dispatch, load allotment and the financial aspect. The OpenAI Gym[19] simulation framework is considered as a widely accepted standard for the development of reinforcement learning (RL) algorithms, and many RL toolkits[18] are designed to adapt to OpenAI Gym. AI4Dist is also compliant with the Gym API standard to allow prototyping using off-the-shelf packages directly.

Interaction with agent modules

The AI4Dist platform is defined to interface with multiple user-defined agents that may have different mechanisms (e.g. learning, optimization, analytic, etc.) controlling various components of the distribution circuit. Every agent has three basic functions:

1. Collect measurements (observations) derived from the current state of the grid.
2. Select actions as a function of present (or also past) observations.
3. Modify the grid component it controls in the environment to reflect the selected action.

At every time step of the simulation, the three functions are automatically called by the environment for every agents in the model to retrieve and implement their control actions. The only information needed by the environment from the agents are (1) what observation they need and (2) the final change in parameters of grid component in simulation. This way the decision-making process is completely isolated from the environment, so users are able to use any external packages or existing software libraries without causing compatibility issues. A template and many examples of agent implementation are provided with the installation of AI4Dist.

Multi-timescale simulation flow

Power system components need to be analyzed and simulated at their appropriate timescale. For example, electricity price usually stay constants for minutes during each market clearance interval, while the response of protective relay only need to be simulated for at most a few seconds after faults. The coordination among agents with different timescales is pivotal to the efficiency of the simulation workflow. In AI4Dist, we divide the simulation to three brackets of timescales: Market, Steady-State and Transient. Agents need to declare their operation speed, as part of one of the brackets above, so that they are called at the right time as the simulation progresses.

The market simulation takes all exogenous pre-determined input files including weather (related to wind and solar output), load profiles and locational marginal price (LMP) from the transmission system. Market participants may be controlled by a prosumer agent that submit market actions including price-curve, bids, offers, etc. depending on type of market mechanism, or be deterministic and produce/consume their default amount at whatever price decided by the market. At the beginning of each market interval, the market collects the market actions from all pro-sumers and computes the LMP at each bus of the circuit. Generators get their scheduled real power output during the whole market interval and flexible loads that contain demand response options receive the capacity they need to curtail as well as the financial incentive for being flexible. During the interval, mismatch between scheduled and actual capacities can be cleared at the LMP of each load. The generator dispatch and load capacity are passed over to the next timescale, the steady-state simulation.

The main purpose of the steady-state module is to implement market result and compute the corresponding power flow solutions. Depending on the resolution of input files, the steady-state power flow can be run multiple times during each market interval. In each solution, the voltage, current and power are obtained throughout the distribution circuit. Agents that work around these measurements can submit their actions after every power flow solution. For example, active voltage regulation algorithm for capacitors, voltage regulators and inverters can adjust their active power output. Each power flow solution serves as the initial condition for potential disturbances and transient events that follow.

For disturbances such as faults, lightning surges or sudden loss of load and generators, transient analysis is necessary for the evaluation of grid stability and equipment effectiveness. Users can input a list of timestamped disturbance events prior to simulation or instruct the environment to randomly create them with specified rate and range of events. Immediately after a disturbance occurs in the circuit, a transient simulation is initialized using the closest available power flow solution. The time-step of this simulation can be adjusted but is usually set at the millisecond level. At each time step, the newest value of voltage, current and power is calculated and provided to agents that are set to operate during disturbances. Typical agents including protective devices (fuses, recloser/breaker relay, etc.) and generator/inverter control loops. The simulation is continued until the transient component has decayed or the circuit is forced to shut down due to cascading failures or oscillations. If the circuit is able to enter a new steady-state with a different topology, caused by the disturbance, the new state will be preserved in the following steady-state simulation.

1.3 Agent module implementation

In this section, we explain the simulation workflow of AI4Dist in detail and give instructions and examples on how to build an agent using any customized algorithm and how to integrate it with the simulation platform. By default, the environment attempts to call certain private functions for each agent object at various stages of the simulation. These functions need to be implemented by users and are expected to return values or update the values of some fields in the agent object they belong to. Every agent needs to have these functions defined before attempting to run a simulation (even though their definitions can simply be empty or return nothing, i.e. null actions). As a minimally working example, an agent should at least implement the following two functions: (1) the function *observe()* to get observation from the environment, and (2) *set_action()* that updates the parameters of the grid component which is controlled by the agent.

Figure 4 shows the cyclical data flow during the simulation process. At the beginning of simulation and after obtaining each

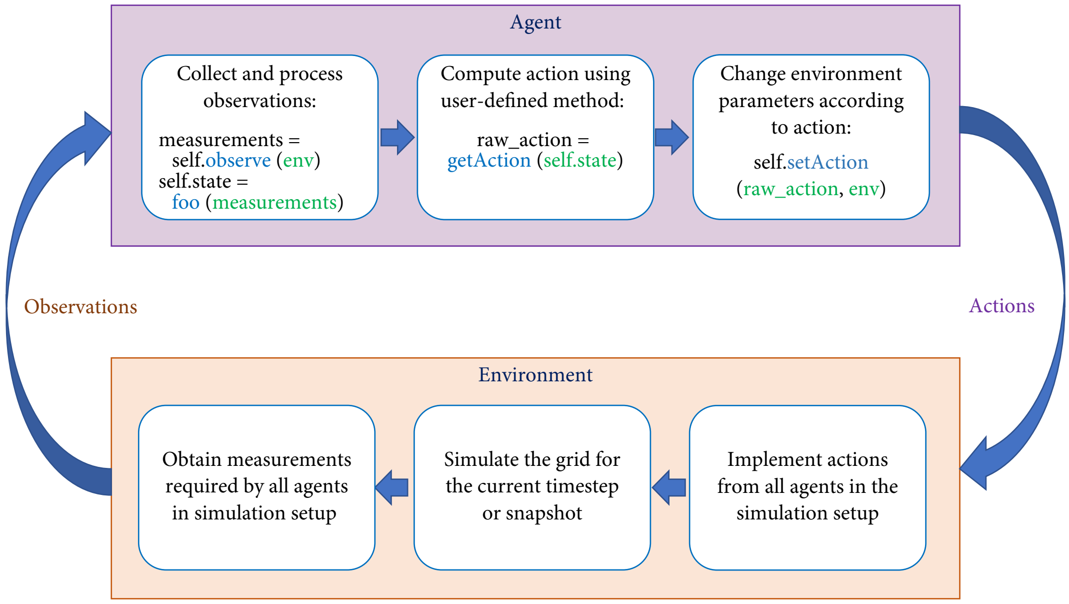


Fig. 4 Interaction between agent and environment.

solution, The environment goes through all agents that are set to operate at the current timescale by calling the *observe()* function and let agents obtain the grid measurements they need. Agents are also free to keep any number of type of internal states in case they are needed. The most critical part of an agent is the mapping from states and observations to actions. This function will be called after all observation routines have finished. In a simple Markovian model, the action can be determined solely from the updated states, but the user is also free to include external data if needed. For machine-learning based models, the raw action obtained thorough model will most likely be a combination of encoded values, which then need to be interpreted and converted to parameter changes in the corresponding grid component. It would be cleaner to separate the model itself and the function that interpret and implement actions, as the latter is mostly of uniform format across all different agents. Table 1 shows a list of basic method functions required for every agents that will be automatically called during the simulation workflow.

A special module beside standardized agents that can also be customized by users is the market mechanism. Similarly to the agents implementation, each market is also an independent Python object that inherits from a provided *abstract class* in the market module. A market object sets the dispatch of generators, loads and nodal price using the supply and demand information provided by generators and loads. We provide three examples of market mechanisms with AI4Dist: (1) Fully regulated distribution in which the price for generation and consumption across the entire circuit is fixed and distributed generators are given the same price as loads' energy price; (2) An optimal power flow (OPF) based market, in which generators and loads submit fanxiexian_myfh/kWh offers and bids, then the market runs security-constrained OPF to determine the price distribution and dispatch; (3) A peer-to-peer (P2P) transaction based market, in which distributed generators and loads are individually paired by their supply and consumption capacity and negotiate a private price between peers using game-theoretic approach.

1.4 Run-time benchmark

The AI4Dist platform along with its simulation engine is able to effectively simulate distribution networks with different scales in a multi-agent setting. Note that the total simulation time of experiments heavily relies on the quantity and type of agents. The training of machine-learning or data-science based agents can take the majority of simulation time even the physical simulation of network model in the platform is relatively fast. To provide a benchmark of the computation speed of the platform, we test and document the time taken to run episodes of transient simulation with 5 naive agents that execute actions according to simple logical expression. Each episode simulates 10 seconds of transient with a time-step length of 1 cycle (16.7 ms). The test is conducted on a

Table 1 Standard function names for agent objects

Name	Description
initialize()	Pre-allocate containers and get necessary grid information before simulation (optional)
observe()	Collect required measurement values from the environment
getAction()	Compute the action using the newest observations
setAction()	Adjust component parameters in environment to match the new actions
reset()	Re-initialize internal states before each episode of simulation for stateful agents (optional)
build()	Initialize a handle to external model or neural-network of given dimension (optional)
train()	The routine to train the agent before running simulation (learning-based agents only)
load()	Load previously stored model (learning-based agents only)
save()	Save weights of trained model to a provided path (learning-based agents only)

mid-range desktop computer with 16 GB of RAM, AMD Ryzen 5 3600 and Samsung 970-evo SSD and the results are listed in Table 2. Initialization time refers to the time taken for this platform to read case files (including topology, bus positions, connection, load, etc.).

Table 2 Simulation time of network models with different sizes

Network size	Initialization time (s)	Simulation time (s)
34	0.163	0.112
123	0.209	0.178
772	0.352	0.830
1692	0.441	1.597
5240	1.246	5.696

2 Case studies

In this section, we provide two concrete examples to demonstrate the capability of AI4Dist on potential use cases. Each case study requires the use of a combination of different agent types and simulation timescales. The code and Jupyter notebooks for demonstration of all case studies will be available on the GitHub repository.

2.1 Case 1: adaptive protective relay

In conventional protection design for distribution systems, the network is often considered to be radial with only one power supply which represents the substation that pulls power from the transmission grid. The detection of faults using current magnitude and the coordination between protection devices (breaker/recloser relay, fuse, sectionalizers) is developed based on this radial assumption which implies one-way power flow from source to loads in every branch. However, the increasing penetration of DER may render this assumption invalid by creating backward power flow during both normal and fault conditions. The fault current contribution from DERs can severely disrupt the reliability and selectivity of conventional protection system by reducing the fault current measured at the source and smaller than the fault current measured near the fault point. For future distribution with many DER and more volatile load profiles, a robust protection design that is not affected by the new characteristics of distribution system operation will be necessary.

We test the performance of several data science and AI based methods on AI4Dist and compare them with conventional inverse-time overcurrent protection. Each agent is placed at the substation, receive local voltage and current measurements during transient simulation and output the fault flag signal to a recloser at the same bus. The measurements are generated through running a multi-timescale simulation that starts with weather and load data, which is used to compute a series of minute-level power flow snapshots of the feeder circuit. Fault scenarios are created by randomly adding disturbances to the system based on initial conditions defined by the snapshots. Figure 5 shows the current waveform from 10 days of continuous simulations and a fault scenario created through transient simulation based on one snapshot.

During the fault scenarios, the agents are expected to initiate a reclosing sequence and try to clear transient faults as quickly as possible before any fuses in the circuit melts, as long as the fault is within the agents' designated operational region. A total of 4 different fault detection methods: overcurrent, support-vector-machine, deep neural network and reinforcement are implemented

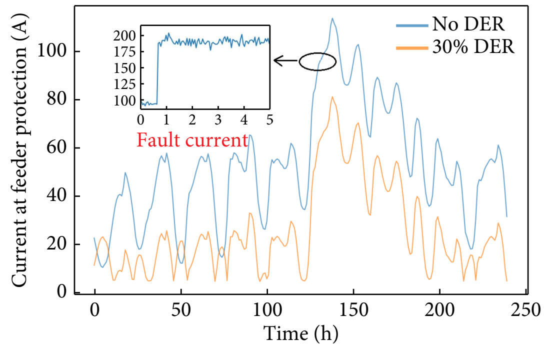


Fig. 5 Feeder source current at different timescales.

as agent and tested in several circuits from the synthetic Austin low-voltage distribution feeders[20]. AI4Dist also provides many handy tools for automation, visualization and analysis of result. Circuit information is automatically read, sorted and converted to accessable format upon initialization. An example is shown in Figure 6 to demonstrate the visualization capability. Under the plug-and-play framework of AI4Dist, implementing and testing agents that have vastly different mechanisms is very simple, as in this example, the user only need to modify the function that compute an output flag from the incoming measurement to switch to a different algorithm.

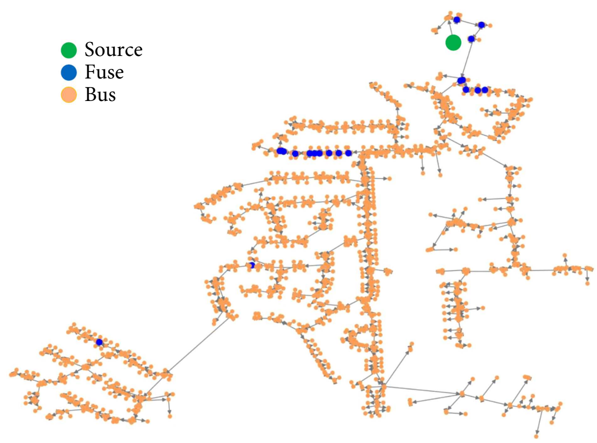


Fig. 6 Example configuration of one synthetic Austin distribution feeders.

2.2 Case 2: reinforcement learning-based voltage support from PV inverters

Most DER and renewable generation require inverter interfacing to connect to the AC grid. Conventionally, many inverter controllers are programmed to produce the maximum real power at 1.0 power factor regardless of the grid operating condition. However, the newly revised IEEE Std. 1547-2018 requires DERs to provide voltage regulation support when they are connected to the grid. Although the reactive power adjustment of a standalone inverter is a trivial task, it becomes more complex when there are many grid-connected DERs that have different power factor settings, speeds and feedback loops for voltage regulation. Conflicting control could result in voltage ripple and create unwanted oscillations throughout the whole distribution circuit. Hence, a reliable control scheme is needed to maintain power quality under DER-heavy grids.

We design a case study based on the DER reactive power support problem using a 240-bus distribution system[21] that is based on a real circuit in the Midwest region of the USA, used to test the performance of agents that control the inverters. This case effectively demonstrate scenarios in which many external agents work in the same simulation setting to achieve an unified objective as shown in Figure 7. User-defined agents are assigned to each DER in the circuit to control their reactive power output while maintaining the apparent power that is determined by weather profile. The design of AI4Dist enables the use of decentralized control for agents and they collectively learn to develop a de-centralized policy that stabilizes the voltage profile in the circuit for different load and DER output capacities based on real profile data.

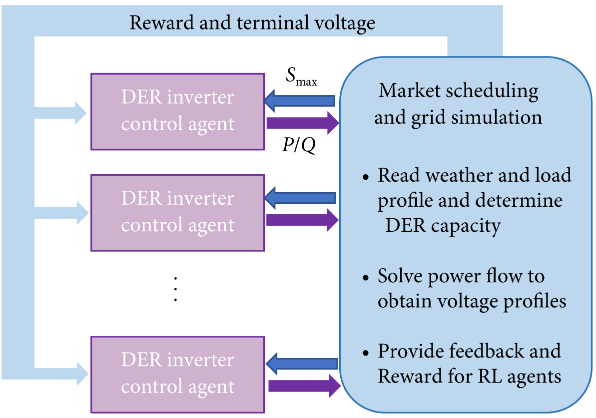
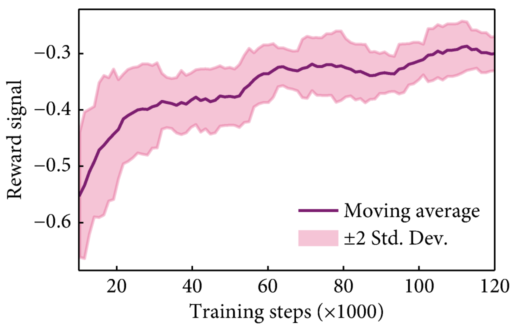
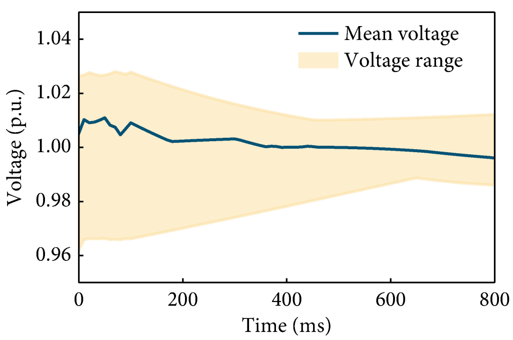


Fig. 7 Agent and environment interaction for voltage support control.

To illustrate the capabilities of AI4Dist, we implement the fully decentralized RL-based control proposed in ref. [5] on the same 240-bus distribution system[21]. Agents were trained to collectively maximize the same reward function, in attempt to regulate voltage. Figure 8 illustrates how the collective average rewards increase during the training process (top), and how the voltage is driven closer to nominal (bottom).



(a) Average rewards during training



(b) Voltage time-series using trained agents

Fig. 8 Sample results from Case 2.

Note that the simulation result is shown in Figure 8 is performed at a relatively fast timescale compared to that of the market. Namely, each time step in this simulation is on the order of 10 ms, whereas market clearance resolution is on the order of 10 min. To account for the coupling between both the physical voltage regulation process and the market clearing process, the voltage regulation process receives the slow-time-scale dispatch from the market as initial conditions at every market clearance. As a result, every 60, 000 steps or so, the initial conditions of the fast-time-scale simulation are updated, and the power flow solutions are recalculated accordingly.

3 Concluding remarks

This paper presents a new power distribution simulation and analysis platform, AI4Dist. It offers several features that are desirable in future distribution grid operation with many DERs. The platform allows easy integration of state-of-the-art data science and AI tools to be integrated with power distribution simulation. It also allows for a cross-time-scale simulation ranging from electromechanical transients to market operations. Example implementations and case studies are illustrated to demonstrate the capability of this platform. AI4Dist could serve as a convenient and powerful tool for developing, testing and benchmarking innovative algorithms for the control and monitoring of modern and future distribution systems.

Building upon AI4Dist, the research community can further extend its capability to include more timescales (e.g. electromagnetic transient). We plan to work on an extension to realistically model uncertainties in distribution grid operations that comes from weather impacts and human factors at demand side, as the current version relies on exogenous inputs as baselines. We will also implement more examples of agents and market mechanisms to assist users in bootstrapping their own studies.

Acknowledgements

This work is supported in part by the Breakthrough Energy Sciences, and in part by the Texas A&M Engineering Experiment Station Smart Grid Center.

Article history

Received: 1 January 2022; Revised: 9 March 2022; Accepted: 10 March 2022

Additional information

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Tesla (2021). Wall connector. Available at https://www.tesla.com/support/home-charging-installation/wall-connector.
[2] Toubeau, J. F., Bottieau, J., Vallée, F., de Grève, Z. (2019). Deep learning-based multivariate probabilistic forecasting for short-term scheduling in power markets. *IEEE Transactions on Power Systems*, 34: 1203–1215.

[3] Cao, Z. J., Wan, C., Zhang, Z. J., Li, F. R., Song, Y. H. (2020). Hybrid ensemble deep learning for deterministic and probabilistic low-voltage load forecasting. *IEEE Transactions on Power Systems*, 35: 1881–1897.
[4] Mandal, P., Senjyu, T., Urasaki, N., Funabashi, T., Srivastava, A. K. (2007). A novel approach to forecast electricity price for PJM using neural network and similar days method. *IEEE Transactions on Power Systems*, 22: 2058–2065.
[5] El Helou, R., Kalathil, D., Xie, L. (2021). Fully decentralized reinforcement learning-based control of photovoltaics in distribution grids for joint provision of real and reactive power. *IEEE Open Access Journal of Power and Energy*, 8: 175–185.
[6] Huang, T., Gao, S. C., Long, X., Xie, L. (2021). A neural Lyapunov approach to transient stability assessment in interconnected microgrids. In: Proceedings of the 54th Hawaii International Conference on System Sciences, Honolulu, HI, USA.
[7] Wu, D. Q., Zheng, X. T., Kalathil, D., Xie, L. (2019). Nested reinforcement learning based control for protective relays in power distribution systems. In: Proceedings of the 2019 IEEE 58th Conference on Decision and Control, Nice, France.
[8] Bahrami, S., Chen, Y. C., Wong, V. W. S. (2021). Deep reinforcement learning for demand response in distribution networks. *IEEE Transactions on Smart Grid*, 12: 1496–1506.
[9] Ji, Y. T., Buechler, E., Rajagopal, R. (2020). Data-driven load modeling and forecasting of residential appliances. *IEEE Transactions on Smart Grid*, 11: 2652–2661.
[10] Dugan, R. C., Montenegro, D. (2018). The open distribution system simulator™ (openDSS), reference guide. Available at https://sourceforge.net/p/electricdss/code/HEAD/tree/trunk/\Distrib/Doc/OpenDSSManual.pdf.
[11] Palmintier, B., Krishnamurthy, D., Top, P., Smith, S., Daily, J., Fuller, J. (2017). Design of the HELICS high-performance transmission-distribution-communication-market co-simulation framework. In: Proceedings of the 2017 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES), Pittsburgh, PA, USA.
[12] Xu, Y., Sun, W., Qu, Z. H. (2021). Renewable energy integration and system operation challenge: Control and optimization of millions of devices. In New Technologies for Power System Operation and Analysis. Jiang, H., Zhang, Y., Muljadi, E. Eds. Waltham, MA, USA: Academic Press.
[13] Blonsky, M., Padullaparti, H. V., Ding, F., Veda, S., USDOE Office of Electricity (2021). OpenDSS-wrapper (distribution system co-simulator with distributed energy resource controls). Available at https://www.osti.gov//servlets/purl/1798964.
[14] Li, H., Tesfatsion, L. (2009). Development of open source software for power market research: The AMES test bed. *The Journal of Energy Markets*, 2: 111–128.
[15] Marot, A., Guyon, I., Donnot, B., Dulac-Arnold, G., Panciatici, P., Awad, M., O' Sullivan, A., Kelly, A., Hampel-Arias, Z. (2020). L2RPN: Learning to run a power network in a sustainable world NeurIPS2020 challenge design. Available at https://www.public.asu.edu/~yweng2/Tutorial5/pdf/111.pdf.
[16] El Helou, R., Lee, K., Wu, D., Xie, L., Shakkottai, S., Subramanian, V. (2022). OpenGridGym: An open source AI-friendly toolkit for distribution market simulation. arXiv preprint, 2203.04410.
[17] Wu, D., El Helou, R., Xie, L. AI4Dist–AI Friendly Power Distribution System Simulation Platform. Available at https://github.com/tamu-engineering-research/AI4Dist----AI-Friendly-Power-Distribution-System-Simulation-Platform.
[18] Raffin, A., Hill, A., Ernestus, M., Gleave, A., Kanervisto, A., Dormann, N. (2019). Stable baselines3. Available at https://github.com/DLR-RM/stable-baselines3.
[19] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W. (2016). OpenAI gym. arXiv preprint arXiv: 1606.01540.
[20] Li, H. Y., Wert, J. L., Birchfield, A. B., Overbye, T. J., Roman, T. G. S., Domingo, C. M., Marcos, F. E. P., Martinez, P. D., Elgindy, T., Palmintier, B. (2020). Building highly detailed synthetic electric grid data sets for combined transmission and distribution systems. *IEEE Open Access Journal of Power and Energy*, 7: 478–488.
[21] Bu, F. K., Yuan, Y. X., Wang, Z. Y., Dehghanpour, K., Kimber, A. (2019). A time-series distribution test system based on real utility data. In: Proceedings of the North American Power Symposium (NAPS), Wichita, KS, USA.