sciendo

B&H Electrical Engineering, Volume 16, Issue 1, 2023:26-34
ISSN:2566-3143, eISSN:2566-3151, DOI: 10.2478/bhee-2023-0004

Review scientific paper

Submitted: March 27, 2023

Accepted: May 16, 2023

REINFORCEMENT LEARNING IN POWER SYSTEM CONTROL AND OPTIMIZATION

Alen Bernadić[1,4], Goran Kujundžić[2], Ivana Primorac[3]

Abstract: Reinforcement learning (RL) is area of Machine Learning (ML) and part of wide-range portfolio of the Artificial intelligence (AI) methods. Besides the explanations of the concepts and principles of RL, in the paper are presented practical RL models for control and optimizing operation of power system – controlling tap-changers for maintain voltage levels and model for techno-economical optimizing operation of energy storages of households in microgrid. Trained RL agent in the practical example synchronizes operation of tap-changers to maintain satisfactory voltage level for the consumers, even in the network with distributed generation. Energy storages are in wide use in households, especially in the combination with PV. In the second example, microgrid's energy management system (EMS) RL agent after learning process act in the simulated environment with variable electrical energy prices, variable load profiles and efficiency of PV modules of households to maximize profit for the houseowners in the microgrid. Agent controls charging and discharging of energy storages and obtain maximal benefit in randomly determined conditions of microgrid operation and different tariff situations. Models are implemented in the Python programming environment Python with specialized power system simulation software (Pandapower) and RL libraries (RLib, OpenAI).

Keywords: Artificial Intelligence, Reinforcement learning, power system, microgrid, Python

INTRODUCTION

Usage of Deep learning (DL), Reinforcement learning (RL), Computer Vision (CV) and other AI methods is known for relatively long period in the power systems [1]-[3]. In this work, reinforcement learning model is tested and implemented on simulated power system environment in the Python ecosystem. Presented application is a prototype of controlling energy management systems (EMS) for the microgrids. It's important to point on applicability of RL methods in different engineering and scientific areas. RL is applied in the robotics, space engineering and in this moment is the mainstream of scientific community. Appliance of RL is part of efforts for increasing of efficiency and reliability of power systems in the days when demands for highest availability is the imperative in operation of contemporary power electricity networks. Optimization of power system operation and minimizing costs are further tasks for AI, which is new paradigm in usage and controlling of energy systems.
Main contribution of the paper is practical implementation of DDPG reinforcement learning algorithm with continuous variables, which are intrinsic for continuous processes, on power systems. Presented approach is contribution to building modular, vectorized and easily upgradeable AI models for controlling and optimization operation of modern and fast changing power systems. The paper is organized as follows: The first section contains information about RL and a literature survey of RL applied in the power systems. The second section brings two practical RL models for optimization power system operation implemented in Python-based RL tools: Techno-economical optimization of energy storages inside the microgrids and simultaneous controlling of tap-changers in power system with renewables. The third section gives direction for future research followed by the Conclusion section.

1. REINFORCEMENT LEARNING

Reinforcement learning (RL) is a branch of machine learning that is concerned with how an agent (The agent could be software module, robot arm, chess player etc.) can learn to make a sequence of decisions in an environment to maximize some notion of cumulative reward. Agent interacts with the environment by taking actions and observing the resulting state and reward. The

[1] Elektroprijenos BiH, Bosnia and Herzegovina
[2] Hrvatske telekomunikacije d.d. Mostar, Bosnia and Herzegovina
[3] University of Rijeka, Croatia
[4] Correspondence email: alen.bernadic@elprenos.ba

goal of the agent is to learn a policy, which is a mapping from states to actions, that maximizes the expected cumulative reward over time. At each time step, the agent observes the current state of the environment and selects an action to take. The environment then takes transitions to a new state according to the transition probabilities, and the agent receives a reward based on the new state and action. The environment sends reward (numerical value) and new environment state of agent for agent's memory for selecting actions with more benefit. Agent's final aim is to maximize the total reward. Generic RL process is described with Figure 1.

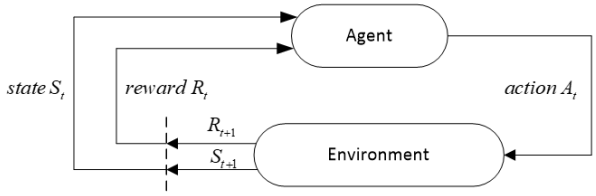

Figure 1. Generic RL algorithm

In RL, the environment is typically modelled as a Markov Decision Process (MDP), which consists of a set of states, actions, transition probabilities, and rewards. Bellman equation is a theoretical basis of reinforcement learning algorithms [4]. According to it, value of the state-action pair in current time-step is equal to the reward r in current time-step plus reward value in the next step discounted with coefficient γ, following policy π as stated in (1)

$$R_t = r_{t+1} + r_{t+2} + ... + = \sum_{i=t}^{T} \gamma^{i-t} r_i \tag{1}$$

Policy, in the RL algorithm denoted with π, define agent behaviour inside the environment i.e., define which action agent should carry out in every possible state of the environment. Optimal policy leads the agent toward maximal reward. The goal of the agent is to learn a policy that maximizes the expected cumulative reward over time. This is typically done by iteratively improving the policy using methods such as value iteration, policy iteration, Q-learning, or actor-critic methods. These methods update the policy based on the observed state, action, reward, and resulting state. RL has applications in a wide range of domains, including robotics, recommendation systems, and autonomous vehicles. One of the advantages of RL is that it can learn to make decisions in complex, uncertain environments without requiring a large amount of labelled data.

***Q* function** is a value function of state-action pair (s,a), which evaluate reward agent gains if in state s take action a following policy π. Value of the state-action pair (or Q-value) is denoted with $Q(s,a)$:

$$Q^{\pi}(s,a) = \left[R(\tau) | s_0 = s, a_0 = a \right] \tag{2}$$

Bellman equation is the theoretical basis of RL [1]. According to it, instantaneous value of state and action is equal to instantaneous reward in observed state with taken action and discounted (γ) reward for the next time step of algorithm, following policy π. Bellman equation can be expressed with Q-value and probability of transition from state s in state s' (P) as:

$$Q(s,a) = R(s,a) + \gamma \sum_{s' \in S} P(s'|s) V(s') \tag{3}$$

with

$$R(\tau) = \gamma_0 r_0 + \gamma_1 r_1 + ... + \gamma_n r_n \tag{4}$$

Where P denotes probability of transition from state s' to the state s and γ is discount factor that defines which rewards are agent's priority – rewards which gains benefit immediately or few steps later. Q function gives expected reward which agent would achieve starting from state s and taking the action a following determined policy. Expected gain calls Q-function or Q- value.

Episode – Duration of execution RL algorithms theoretically can be infinite for continuous processes as stock markets or power system operations. However, discretization of observed processes or its parts is better way for detailed approach – it's discrete processes i.e. 24 – hours operation of microgrid or 60 – seconds operation of control relay. Therefore, an episode consists of finite time steps is defined along with the definition of episode's end.

1.1. Reinforcement learning in the power systems – a literature survey.

Reinforcement learning (RL) has been applied to various aspects of power system networks, including control, optimization, and management. Main RL applications in power system networks are:

Power grid control: Power flow control, Voltage control and Frequency control: RL can be used to control the power flow in the transmission network by adjusting the settings of the power system devices, such as transformers and capacitors, voltage regulators and reactive power devices and parameters in protection and control relays. RL algorithms can learn to control the devices' settings to optimize the power flow in the transmission network, ensuring that the power demand is met, and the network remains stable [5], [6].

Energy management systems (EMS): RL can be used to optimize energy management systems in power networks, including demand response, energy storage, and renewable energy integration. RL can learn to control these systems by maximizing the energy efficiency and reducing energy costs. Main RL applications in EMS are Energy storage [7], Renewable energy integration [8], Energy trading [9] and Energy efficiency [10]-[12].

B&H Electrical Engineering, Volume 16, Issue 1, 2023:26-34
ISSN:2566-3143, eISSN:2566-3151, DOI: 10.2478/bhee-2023-0004

Fault detection and diagnosis: RL can be used to detect and diagnose faults in power systems, such as transmission line faults, transformer faults, and generator faults. RL algorithms can learn to identify the root cause of faults and provide recommendations for corrective actions. Some examples of RL applications in fault detection are Transmission line, Transformer and Generator fault detection: RL can be used to detect power system element fault, such as short-circuits. RL algorithms can learn to analyse the real-time power system data and identify the abnormal patterns that indicate the presence of a fault [13]-[16]. RL can improve fault detection and diagnosis in power systems by enabling early detection of faults, reducing the downtime and maintenance costs, and improving the power system reliability and safety.

Load forecasting: RL can be used to forecast the load demand in power systems. RL algorithms can learn to analyse historical load data and external factors such as weather patterns and events to predict the future load demand accurately. Main RL applications in load forecasting are Short-term load forecasting [17]-[18], Renewable energy forecasting [19] and Energy market forecasting [20].

Power system scheduling: RL also can be used to optimize the scheduling of power generation and transmission. RL algorithms can learn to optimize the scheduling of generators, transmission lines, and energy storage systems by maximizing the efficiency of the system and minimizing the operational costs. Main RL applications in power system scheduling are Unit commitment [21], Economic dispatch [22], Reserve scheduling [23] and Demand response [24].

Overall conclusion is that appliance of practical RL algorithms can improve power grid control by enabling efficient and intelligent decision-making, resulting in improved system stability, reliability, and cost-effectiveness of power systems.

1.2. Selection of RL algorithm for the practical examples

Practical examples of RL in the power systems demand work with continuous variables in action space (change tap position of transformer regulators, charge/discharge batteries in households - amounts in kW) and observation (State of charge in batteries, hourly prices in €/kWh etc.) spaces as explained in next sections. Handling with vector of continuous variables (controlling three energy storages in the microgrid) is intrinsic for Deep Deterministic Policy Gradient (DDPG) algorithm [25], a variant of the most effective RL Actor-critic algorithm, so it is chosen for implementation of practical algorithm in the Python ecosystem. Furthermore, chosen algorithm is model-free and designed for environments with continuous variables (continuous action and observation spaces), which is suitable for power systems RL applications. DDPG calculates policy on deterministic way, meaning that policy define exact action for every agent state in the environment, noted as . It combines the actor-critic architecture with the deep neural networks to learn a deterministic policy that maps states to actions. The algorithm is based on the idea of using the policy gradient method to update the actor network and using the Q-value function to update the critic network.

The DDPG algorithm works by training four neural networks, main approximator and target neural network for actor and critic subsystem [25]. The actor network system finds the optimal policy for the agent and critic network takes in the current state and the action taken as input and outputs the expected cumulative reward for taking that action in that state as presented on Figure 2.

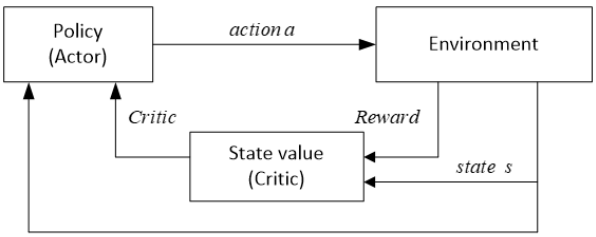

Figure 2. Structure of the Actor-critic algorithm

Action a of actor network is determined with on-policy algorithm (policy gradient) which optimizing agent's policy. DDPG algorithm by its definition is algorithm with deterministic policy meaning that gives action for exact state s of the environment. Parameter of actor neural network is denoted with ϕ. Such determined action is input into the critic network along with state s. Critic network as output have Q-value, which makes principal scheme of Critic network as on Figure 3

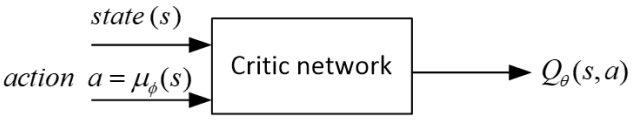

Figure 3. The critic network scheme

Critic networks evaluate quality of agent's actions (value network) approximated with actor neural network and calculate Q function. Q function gives expected reward which agent realize starting from state s taking an action a following certain policy as in equation 3. Gain realized with Q-function is called Q value. Thus, for the action a undertaken in state s we obtain Q-value, which is measure of agent's action quality in current state. Critic network uses deep neural networks or Deep Q networks (DQN) as value approximator of Q – values which is produced by the Actor network $Q_\theta(s,y)$, with θ as neural network parameter in the critic network. Generally, neural networks learn with minimizing the difference between target and approximated (predicted) values by correction of neural networks parameters. Target value of DQN network is

optimal value, which is calculated by Bellman equation. Aimed Q-value of critic's neural network can be calculated as:

$$Q^*(s,a) = r + \gamma \cdot \max_{a'} Q(s',a') \tag{5}$$

Therefore, loss function L of critic's neural network can be expressed as:

$$L(\vartheta) = r + \gamma \cdot \max_{a'} Q_{\vartheta}(s',a') - Q_{\vartheta}(s,a) \tag{6}$$

With $a = \mu_{\theta}(s)$ determined with actor network. Finally, loss function of critic's neural network can be expressed as function of expectation E of transition from state s to s' as:

$$L(\vartheta) = \underset{(s,a,r,s')}{E}\left[(Q_{\theta}(s,a) - (r + \gamma \max_{a'} Q_{\theta}(s',a'))^2\right] \tag{7}$$

To improve stability and convergence, DDPG uses a replay buffer to store experiences and randomly samples batches of experiences to train the networks. It also uses target networks, which are copies of the actor and critic networks that are periodically updated with the weights of the original networks.

2. PRACTICAL EXAMPLES OF RL IN POWER SYSTEM

2.1. Techno-economical optimization of battery operation in the Microgrid

According to [26], Microgrids are small-scale, LV CHP supply networks designed to supply electrical and heat loads for a small community. There are many types of business arrangements for household owners inside of microgrid, but in common all have one aim – optimizing use of equipment and maximizing profit. Energy management systems (EMS) are automation systems that collect energy measurement data from the field and making it available to users through graphics, online monitoring tools, and energy quality analysers, thus enabling the management of energy resources.

For the practical example, an microgrid configuration on Figure4 is chosen. External 20 kV grid is connected with 0.4 kV microgrid via transformer 20/0.4 kV in the point of common couple (PCC) with circuit breaker. Microgrid consist of three households each equipped with PV solar panels and house batteries with basic data denoted on Figure 4. Home batteries up to 50 kWh capacity is a relatively new option for homeowners who have solar panels installed and want to store excess energy generated during the day for use during periods of high energy demand. When the battery is charged, it stores electrical energy that can be used later, and when it is discharged, it releases that energy to power devices or appliances. There are several ways to charge home batteries. One common method is to connect the battery to a solar panel system, which generates electricity and sends it to the battery for storage. Another way is to charge the battery from the electrical grid during off-peak hours, when energy is cheaper, and demand is lower. The battery's capacity is measured in kilowatt-hours (kWh). In chosen example configuration power network, three households forming an microgrid, generates its own solar energy with PV power plants up to 23 kW and can store this energy in a home battery unit with up to 50 kWh capacity. All households have relatively large loads during working hour of day. Microgrid electrical management system of microgrid (EMS) control and synchronize usage of devices depending on tariff, hourly price, hour of sunlight and availability of equipment. The electricity contract of the household is based on day-ahead market prices and the storage device can also be used to participate in a secondary reserve market and make additional benefit for homeowner. For residential use, home batteries typically have much smaller capacities, ranging from a few kWh to a few tens of kWh. These smaller home batteries can still provide significant benefits by allowing homeowners to store excess solar energy generated during the day and use it at night or during times of peak demand, which can help reduce their reliance on the grid, lower their electricity bills and make profitable arrangements with distribution system operator with surplus of energy in battery storages.

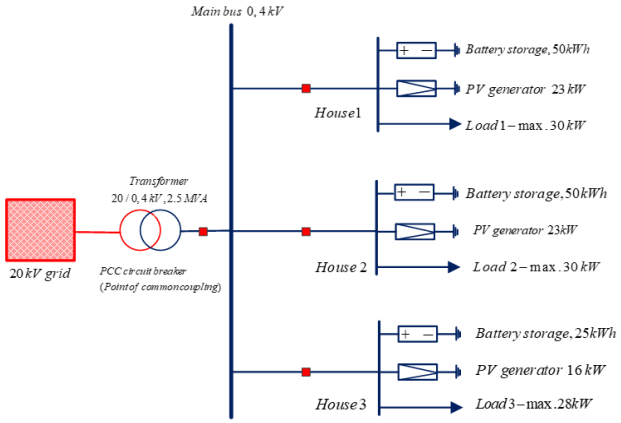

Figure 4: Microgrid configuration

2.1.1. Pandapower Microgrid simulation

For training process of RL algorithm, presented microgrid is modelled in open-source software tool for power system networks simulation - Pandapower [27], [28]. Pandapower provides a range of tools for visualizing and analysing simulation results, including graphs and tables that show the behaviour of the system over time and provides a range of functions for simulating power flow, short-circuit calculations, and other power system analyses, including battery systems. As intrinsically pythonic software, Pandapower is easily integrated in python ecosystem consist of Anaconda python distribution [29] along with AI tools for training and executing

B&H Electrical Engineering, Volume 16, Issue 1, 2023:26-34
ISSN:2566-3143, eISSN:2566-3151, DOI: 10.2478/bhee-2023-0004

RL agents. In that way, microgrid configuration is modelled and calculations in power system are executed simultaneously with RL training process in one software environment. In this project, a simulation environment was built, using the OpenAI Gym [30] as a framework. Especial attention is given to the simulation of battery storage in Pandapower simulation tool. The standard function (pandapower.create_storage) is used to create a storage element. The minimum set of function arguments is following: the bus ID to which the storage element is connected (bus), active and reactive power of the storage element (p_mw, q_mvar). The positive sign of the storage real power reflects the charging process while the negative sign – discharging. The example of the function for the storage creation in Pandapower simulation software is as follows:

storage_index = pp.create_storage(net, bus=5, p_mw=0, max_e_mwh=0.050, soc_percent=50, max_p_mw=0.005, min_p_mw=-0.005)

2.1.2. Simulation parameters, load profiles, PV efficiency and day-ahead prices

Each households have load profile with relatively large consumption in working hours of day, as on Figure 5. Rated power of PV plants is denoted on Figure 4 and instantaneous power is determined with multiplying rated power with efficiency coefficient of PV plant related with daylight (Figure 5). Hourly prices in €/kWh are determined by distribution regulator on day-ahead principle. For authentic simulation, all parameters change randomly in every 24-hour episode in range denoted on Figure 5.

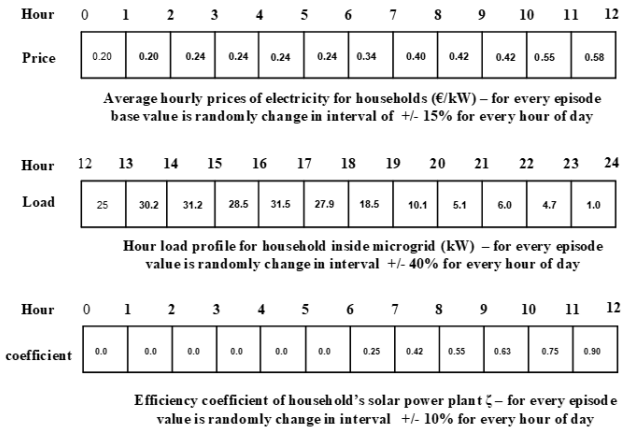

Figure 5: Variable parameters for simulations

Valuable approach and real data for household battery systems are given in [32]. Author describes reinforcement learning methodology and combine it with real data from energy storage and distribution electricity grid. Patterns for household consumption profile and prices are taken from [33]. Presumption is that each households have load profile with relatively large consumption in working hours of day, as on Figure 5. Rated power of PV plants is denoted on Figure 4 and instantaneous power is determined with multiplying rated power with efficiency coefficient depend on time of day and meteorology conditions. All input data are random changeable for better process of EMS agent's training.

2.1.3. Task definition and simulation environment

In the Python simulation environment create Pandapower model of microgrid with three home storage batteries. With reinforcement learning algorithm train an electrical management system (EMS) agent for optimizing operation of microgrid and maximize profit from PV power plants and batteries installed on households inside the observed microgrid. Simulations for training should be conducted with randomly changing parameters of system (household loads, day-ahead electrical energy prices and efficiency of PV solar plants) for each 24-hour episode. Resulting EMS agent should be a controlling system vector with three commands for the microgrid's batteries with charging/discharging amounts of energy and idle state. For continuous amounts of battery charge/discharge energy, and three different functions connected with amounts, for the solution of this task a Deep Deterministic Policy Gradient (DDPG) algorithm is chosen. In reinforcement learning, the agent learns by making actions in an environment, receiving rewards/penalties for it, and modifying its action pattern (policy) accordingly. Our goal is to train an EMS agent that maximizes the profit of a households by controlling the batteries considering the energy markets (day-ahead market, frequency markets), the local conditions (e.g., the baseload of the household) and user preferences (e.g., battery level can't go below certain threshold). Rewarding as part of RL methodology for the chosen actions of RL algorithm is schematically presented with Figure 6 and Figure 7.

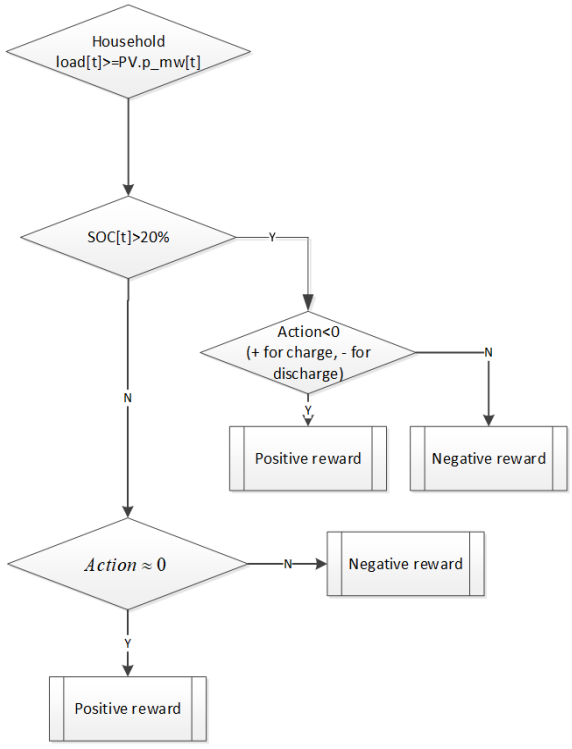

Figure 6: Block diagram for rewarding the battery actions of EMS agent household load bigger than PV production.

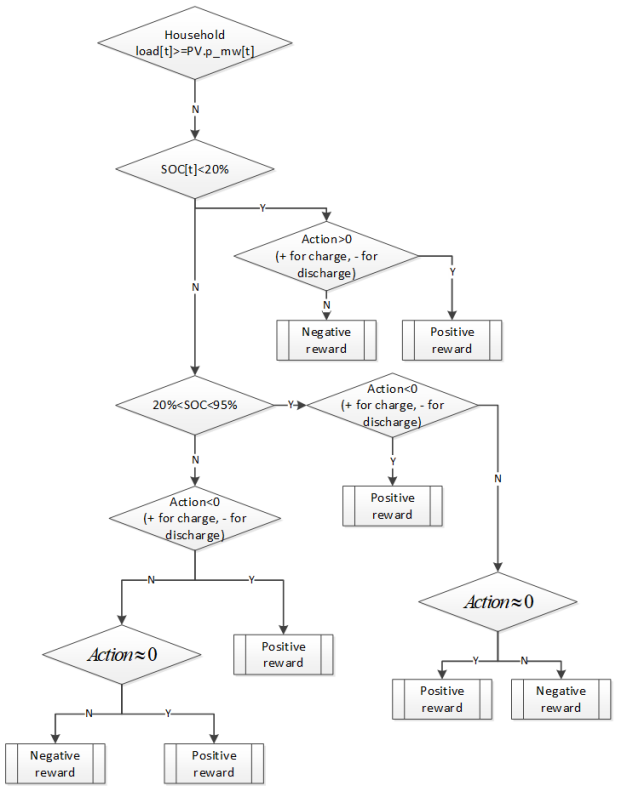

Figure 7: Block diagram for rewarding the battery actions of EMS agent for house load smaller than PV production.

An EMS agent gets positive rewards for right choose of action, quantified with profit in €. Presumption is that household's loads cannot be changed because importance of household's farming or economy. Conditions from Figures 6 and 7 are implemented programmatically in Python RL model. Reinforcement learning algorithms needs defined duration of episode. An episode for selected microgrid configuration is typical 24-hour period in which all parameters change randomly and trained EMS microgrid agent should be able to acquire maximal profit for entire microgrid – all batteries can be controlled individually without mutual restrictions to achieve maximal profit. All simulation must be OpenAI Gym compatible for further compatibility with Stable-baselines library [31] specialized for RL algorithms, which is top module for execution of EMS agent trainings.

2.1.4. Model results

Training process is run on python programming environment on PC with Pentium i5 processor and 8 GB RAM. Duration of training process for 480.000 time steps was approximately 5 hours. Average values for a month of overall microgrid profit are presented on Figure 8. Red curve is profit with microgrid operation lead by train EMS agent and blue curve is microgrid profit with weaker optimized agent. Note: Agent is trained with assumed day-ahead prices with random variation included. For different price profile, approximately the same benefit is obtained with trained EMS agent. It's evident that household profit in microgrid, which is controlled with train reinforcement learning EMS agent, is considerably bigger in comparison with microgrid operation with weaker controlling system.

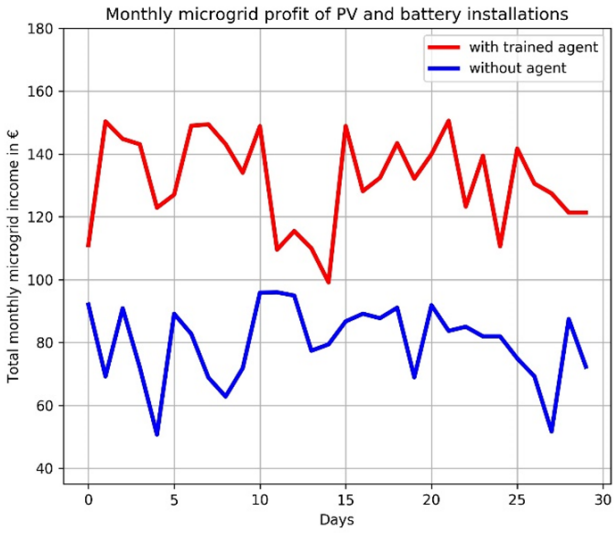

Figure 8: Maximizing profit with RL-trained EMS agent

The DDPG algorithm and corresponding model, developed for starting power system configuration, is also tested with considerably smaller PV plant capacities of households in the microgrid (installed 4kW rated power each with applied efficiency coefficient defined on Figure 5). As expected, income is smaller, but still optimized in comparison with weakly trained EMS agent as presented on Figure 9.

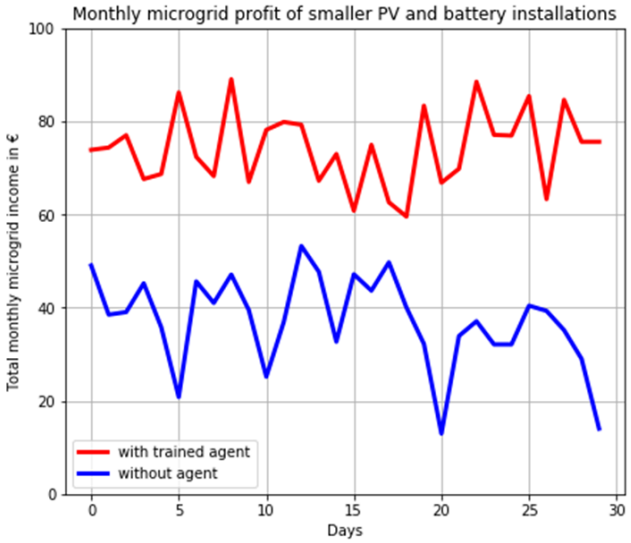

Figure 9: Maximizing profit with RL-trained EMS agent in configuration with smaller PV capacities

In conditions of evidently high prices of the electrical power, optimizing of household's profit inside microgrid or other communal organization is techno-economical imperative for power system distribution systems in future. Maximizing profit from microgrid is elaborated in [34], with valuable and detailed mathematical model for optimizing operation of elements of microgrid. In comparison with methodology

B&H Electrical Engineering, Volume 16, Issue 1, 2023:26-34
ISSN:2566-3143, eISSN:2566-3151, DOI: 10.2478/bhee-2023-0004

in [34], RL agent optimize profit with random hourly input system parameters (household loads, solar radiation, battery state) for each household. Furthermore, RL model is modular and easy extendable for arbitrary number of operational units of microgrid (energy storages, PV plants, controllable loads.) inside the microgrid. Also, RL model is trained in shorter time and gives possibility for programmatically extending rewarding system and optimization with methodology from [34] for further improving of model.

2.2. Synchronized controlling of power system elements.

For the second practical example single diagram of part of the power system is presented on Figure 10. Outer power 110 kV grid and 0.4 kV solar plant are sources for the distribution system. Two transformers 110/20 kV, 40 MVA and 20/0.4 kV, 4MVA are main power system elements in the considered part of electrical distribution network and both are equipped with controllable tap-changers.

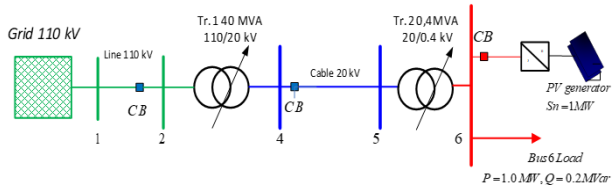

Figure 10: Configuration of the electrical power distribution system for the second example

On the 0.4kV busbar no. 6 is also connected PV solar power plant with 1 MW rated power. Power system is modelled with Pandapower [27]-[28], Python – based simulation software which is easily integrated in overall Anaconda Python distribution [29]. Thereby, power flow calculation and training of RL agents in sequences of predefined episodes are executed simultaneously in the computer system.

2.2.1. Task definition and simulation environment

In the Python simulation environment create Pandapower model of observed part of electrical power distribution system. With reinforcement learning algorithm train a power system controlling agent for optimizing operation of controllable tap-changers for maintaining satisfactory voltage level on the 0.4 kV busbars (between 1.0 and 1.1 p.u. of the rated voltage) for households. Train RL agent must maintain voltage level for variable parameters of the power system. Starting parameters for every episode are randomly set, i.e. main grid 110 kV voltage is in range 0.9 p.u to 1.2 p.u., beginning position of the tap-changer for transformer 110/20 kV is in range -5 to 5 and for transformer 20/0,4 kV in range from -15 to +15. Active and reactive power of PV plant are also randomly set up in realistic ranges. Such randomly chosen parameters results with over or under voltages in the system. Voltage level on busbar 0,4 kV no 6 is quantity of interest and with synchronized manipulation with controllable tap-changers trained RL agent must regulate observed voltage in desired range between 1.0 and 1.1 p.u. of rated voltage for maintain satisfactory power quality of delivered electrical power. Two controlled power system elements make controlling vector of RL agent. Note: For testing of RL algorithm, tap-changers are modelled as controllable for both transformers, which is not the case in practical use of 20/0,4 kV transformers. For the solution of task, reinforcement learning DDPG [25] algorithm is chosen. Action space for the DDPG agent is the vector of commands for tap-changers and injected amount of active and reactive power for PV plant connected on the busbar 6. Agent in the minimum steps must set voltage on the busbar 6 in desired range. By example agent's action vector [-1.0 0 0.08 -0.1] means that RL agent in the current time step lowers tap-changer for transformer 1, do nothing with tap-changer of transformer 2, active power of power plant is augmented for 8% and reactive lowers for 10 % in comparison with previous time-step.

2.2.2. Model results

Algorithm is implemented in the Stable-baselines [31] specialized RL library with DDPG module chosen. Example of results of agent actions are presented on Figure 11 and Figure 12, for initial undervoltage and overvoltage on the busbars 6. Controlling command vector of trained RL agent set up voltage level in the desired range for quality of delivered electrical power. Training process is run on Python programming environment on PC with Pentium i5 processor and 8 GB RAM. Duration of training process for 480.000 time steps was approximately 18 hours.

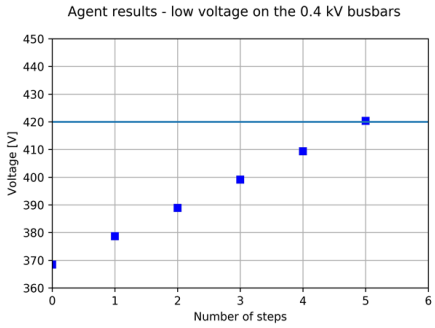

Figure 11: Agent actions for low voltage on busbars 6

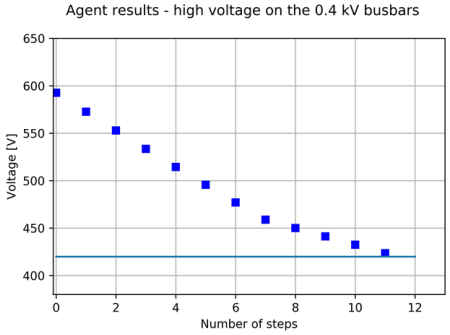

Figure 12: Agent actions for high voltage on busbars 6

It's evident that trained RL agent set up the voltage in desired value with minimum number of synchronized operations with power system elements.

3. FURTHER WORK

Presented model is already extendable and vectorized and it can be applied on controlling more energy storage units in the microgrid. Further development of RL algorithms in the power systems will be in the area of hybrid action-spaces, where agents will be able to control devices with continuous (i.e. battery storages) and discrete variables (e.g. circuit breakers), simultaneously and in the same environment. Such algorithm would be able to perform more complex controlling algorithms for optimizing electrical power grid operation and at the same time increase profit for owners of PV plants, EV and battery storages. For instance, at the same time would be possible to control power quality parameters (frequency, voltage levels...) parameters and in the same execution optimize devices for techno-economic purpose.

4. CONCLUSION

Basic concepts and principles of RL are presented in this work. Practical model of RL agent is implemented with DDPG algorithm in the Python ecosystem. Trained agent can optimize operation of battery storages in the microgrid and maximize household profit. Model is vectorized and extendable for adjusting to bigger microgrids. Presented model and appliance of practical RL algorithms can improve power grid control by enabling efficient and intelligent decision-making, resulting in improved system stability, reliability, and cost-effectiveness of modern power systems.

REFERENCES

[1] R. Aggarwal,Y. Song: Artificial Neural Networks in Power Systems, Part I: General introduction to neural computing, Power Engineering Journal, Volume: 11, Issue: 3, June 1997 , Page(s): 129 – 134.
[2] R. Aggarwal, Y. Song: Artificial Neural Networks in Power Systems, Part II: Types of artificial neural networks Power Engineering Journal Volume 12, Issue 1, February 1998, p. 41 – 47.
[3] S. Khaitan: A Survey Of Techniques for using Neural Networks in Power Systems, https://hal.archives-ouvertes.fr/hal-01631454, 2017.
[4] Sutton, Barto: Reinforcement learning: an introduction, Second ed. Cambridge, MA, 2018.
[5] A. Bernadić, G. Kujundžić, I. Primorac: „Primjena algoritama podržanog učenja u upravljanju elektroenergetskog sustava ", 3. Savjetovanje BH CIRED, Mostar, 2022.
[6] A. Bernadić: „Deep and Reinforcement learning, and Computer Vision Methods in power systems – practical examples in Python ecosystem ", Znanstveno-stručna konferencija: Umjetna inteligencija u BiH/istraživanje, primjena i perspektive razvoja Konferencija / Scientific conference: AI in Bosnia Herzegovina, Intera technological park, Mostar, April 2022., Zbornik radova ISBN 978-9958-11-165-5, Ministarstvo znanosti FBiH.
[7] S. Duque, J. Giraldo, P. Vergara, P. Nguyen, A. van der Molen, H. Slootweg: „Community energy storage operation via reinforcement learning with eligibility traces ", in Electric Power Systems, Research, Volume 212, 2022, ISSN 0378-7796.
[8] Y. Liu et al.: „A Reinforcement Learning-Based Energy Management System for a Hybrid Power System with Renewable Energy Sources ", in International Conference on Power Electronics, Control and Automation (ICPECA), New Delhi, India, 2019, pp. 1-5, doi: 10.1109/ICPECA47973.2019.8975505.
[9] Zang, H.; Kim, J. „Reinforcement Learning Based Peer-to-Peer Energy Trade Management Using Community Energy Storage in Local Energy Market", Energies 2021, 14, 4131. https://doi.org/10.3390/en14144131
[10] S. Kim, H. Lim: „Reinforcement Learning Based Energy Management Algorithm for Smart Energy Buildings", in Energies 2018, 11, 2010. https://doi.org/10.3390/en11082010
[11] K. Mason, S. Grijalva: „A Review of Reinforcement Learning for Autonomous Building Energy Management", 2019, doi: https://arxiv.org/abs/1903.05196
[12] N. Taha, T. Pekka: „Deep RL for Energy Management in a Microgrid with Flexible Demand ", 2020, doi: 10.20944/preprints202010.0156.v1.
[13] M. Li, H. Zhang, T. Ji and Q. H. Wu: "Fault Identification in Power Network Based on Deep Reinforcement Learning," in CSEE Journal of Power and Energy Systems, vol. 8, no. 3, pp. 721-731, May 2022, doi: 10.17775/CSEEJPES.2020.04520.
[14] M. Ibrahim, A. Alsheikh, R. Elhafiz: „Resiliency Assessment of Power Systems Using Deep Reinforcement Learning ", Volume 2022, Article ID 2017366, https://doi.org/10.1155/2022/2017366
[15] M. Glavić, (Deep) Reinforcement learning for electric power system control and related problems: A short review and perspectives, Annual Reviews in Control, Volume 48, 2019, Pages 22-35, https://doi.org/10.1016/j.arcontrol.2019.09.008.
[16] Y. Zhu: „Power Grid Cascading Failure Mitigation by Reinforcement Learning", 2021, https://arxiv.org/abs/2108.10424
[17] Ungureanu, S.; Topa, V.; Cziker, A.: „Deep Learning for Short-Term Load Forecasting ", Industrial Consumer Case Study, Appl. Sci. 2021, 11, 10126.
[18] Yujie Gao et al: „Reinforcement Learning Based Short-Term Load Forecasting with Dynamic Features Selection ", 2021.

B&H Electrical Engineering, Volume 16, Issue 1, 2023:26-34
ISSN:2566-3143, eISSN:2566-3151, DOI: 10.2478/bhee-2023-0004

[19] Daniel Carlos do Vale Ramos: „Reinforcement Learning of a Multi-Agent System for the Forecasting of Electricity Consumption, Dissertation/project report/internship report, University of Porto 2020/2021., available on https://repositorio-berto.up.pt/bitstream/10216/138254/2/519034.pdf, last accessed 16/03/2023.
[20] Lehna, Hoppmann, Heinrich, Scholz: „A Reinforcement Learning Approach for the Continuous Electricity Market of Germany: Trading from the Perspective of a Wind Park Operator “, Fraunhofer Institute for Energy Economics and Energy System Technology (IEE), 2021.
[21] D. Perera, P. Kamalaruban: „Applications of reinforcement learning in energy systems “, Renewable and Sustainable Energy Reviews 137, 2021.
[22] Z. Yu, G. Ruan, X. Wang, G. Zhang, Y. He, H. Zhong: „Evaluation of Look-ahead Economic Dispatch Using Reinforcement Learning“, 2022, doi: https://arxiv.org/pdf/2209.10207.pdf
[23] A. Ajagekar, F. You: „Scheduling of Electrical Power Systems under Uncertainty using Deep Reinforcement Learning“, Computer Aided Chemical Engineering,Elsevier, Volume 49, 2022, Pages 463-468,ISBN 9780323851596
[24] V. Solberg: “Reinforcement learning for grid control in an electric distribution system”, Master thesis, NMBU University, Norway, 2019.
[25] S. Ravichandiran: Deep Reinforcement Learning with Python, Second Edition, Packt Publishing, 2020., ISBN 9781839210686.
[26] S. Chowdhury, S. P. Chowdhury, and P. Crossley: Microgrids and active distribution networks, Institution of Engineering and Technology, 2009.
[27] Pandapower, power system simulation tool (2022), Available: http://www.pandapower.org/,
[28] L. Thurner; A. Scheidler; F. Schäfer: “Pandapower—An Open-Source Python Tool for Convenient Modeling, Analysis, and Optimization of Electric Power Systems”, IEEE Transactions on Power Systems, Volume: 33, Issue: 6, Nov. 2018.
[29] Anaconda Python distribution (2022), Available: www.anaconda.com
[30] Gym reinforcement learning library, (2022), Available: https://www.gymlibrary.dev/
[31] Stable baselines Python RL library (2022), Available: https://stable-baselines.readthedocs.io/en/master/
[32] K.Eljand: „Training an Energy Decision Agent With Reinforcement Learning“,2022., Available: https://towardsdatascience.com/training-an-energy- decision-agent-with-reinforcement-learning-a7567b61d0aa, last accessed 15.4.2023.
[33] EU Energy prices data and visualisation tool, (2023), Available: https://ec.europa.eu, last accessed 11.4.2023.
[34] Geramifar, H., Shahabi, M. and Barforoshi, T. „Coordination of energy storage systems and DR resources for optimal scheduling of microgrids under uncertainties “, IET Renewable Power Generation, (2017), 11: 378-388. https://doi.org/10.1049/iet-rpg.2016.0094

BIOGRAPHIES

Alen Bernadić is employed at Electricity Transmission Company of BiH, as Head of department for planning and engineering of operational area Mostar. He is Assistant professor at Mostar university, Faculty of Mechanical engineering, Computer science and Electrical engineering faculty (FSRE). He is also involved with University of Split as Associate on Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture (FESB). Have published more than 25 scientific papers in international journals and at international conferences. In the latest research special interest is for AI methods in the power system.

Goran Kujundžić received the B.S. and Ph.D. degree from University of Zagreb, Croatia in 2000 and 2017 respectively. He worked as a designer and project manager at the Distribution Department of Elektroprivreda HZ HB power utility (2000-2006) and after at Power Department of JP Hrvatske Telekomunikacije Mostar. His research interests include energy storage systems and management of microgrids that are based on renewable sources.

Ivana Primorac, PhD candidate, was born in Zenica, Bosnia and Herzegovina in 1987. Employed at Electricity Transmission Company of BiH, as Head of the accounting department of operational area Mostar. She is PhD candidate at the Faculty of Economics, University of Rijeka. Have published 2 scientific papers at international conferences. Her research interests include marketing communication and information availability about renewable energy sources.