Implementation details of PPO algorithm for real-time optimization of power flows in electric microgrids, challenges and future directions

Rima OULHAJ [a,*], Pierre GARAMBOIS [a], Lionel ROUCOULES [a]

[a] **Arts et Metiers Institut de Technologie, HESAM Université, LISPEN, 13617 Aix-en-Provence, France**
[*] rima.oulhaj@ensam.eu

1 Introduction

Microgrids (MGs) have allowed distributed energy integration, energy self-sufficiency improvement and renewable energy penetration through the use of energy storage systems and independent management and decision-making processes. Indeed, local management of energy production and storage technologies can have a positive impact on economic interests (e.g., lowering energy costs), environmental interests (e.g., lowering carbon print) and societal interests (e.g., ensuring energy fairness). Energy management in microgrids, which includes power flow management, enables the optimization of these interests while ensuring good energy supply quality. Power flow optimization includes scheduling and real-time scenarios. In this article, we focus on the real-time problem and we implement a policy-gradient deep reinforcement algorithm called Proximal Policy Optimization (PPO) [1] to solve it. The aim of this article is to present implementation details that are related to the specific environment that we're working with (the electric microgrid) in an attempt to ensure implementability of our work and better interpretation of results. We base our work on PPO algorithm optimization recommendations presented in [2] which have become a reference in the community.

2 Context

Energy self-sufficiency has become a pressing issue within the current context (fig. 2.1). Indeed, climate change induces uncertainty of renewable generation (RenGen) including hydroelectric, wind, solar, biomass, geothermic and wave generation [3]. Also, energy supply shortfall in traditional, big-scale power plants (namely due to the lack of maintenance) affects self-sufficiency and (as a consequence) induces variability in energy markets [4]. MGs are a flexible alternative

to the traditional nationwide utility grids (UGs) as they allow independent decision making and are specifically designed and sized to satisfy the power loads of a given area (neighborhood, city, region, etc.). In this context, power flow control in MGs (in both the scheduling and real-time scenarios) has proven itself to be a powerful tool to optimize self-sufficiency as well as any other criteria chosen by the stakeholders (e.g., environmental, economic, etc.). Optimization of power flows must account for the many uncertain signals, namely power demand, meteorological phenomena and UG energy costs. In the real-time scenario, the problem isn't only to fit the control strategy to given 'typical' demand and uncontrollable RenGen profiles, but to be able to adapt and recover from any unpredicted events, such as sudden demand peaks or RenGen low points.

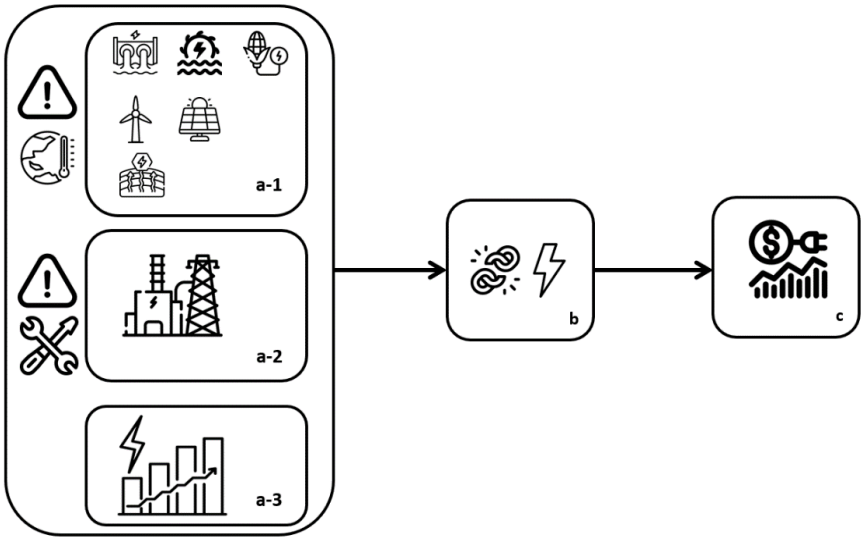

Figure 2.1: Causal connection between uncertainty of renewable generation due to climate change (a-1), insufficient power supply of traditional power plants due to lack of maintenance (a-2), power demand fluctuations (a-3), energy sufficiency (b) and energy markets (c)

3 The real-time optimal power flows problem

In the real-time optimal power flows problem, the objective is to implement an optimization method that adapts to the real-time signals such as the power load, UG energy cost and uncontrollable renewable generation. The optimization method shouldn't focus solely on the immediate outcome of the choice of power flows in the MG, but rather on both the immediate and future outcomes. In section 2.2 we present the optimization problem and in section 2.3 we formalize it as a Markov Decision Process (MDP).

3.1 Optimization problem : decision variables and constraints

We consider a MG where controllable technologies are of two types: controllable generators (CGs) and energy storage systems (ESSs). For each CG, let $\boldsymbol{x_t^g}$ be the desired power output from generator $\boldsymbol{g}$ at time step $\boldsymbol{t}$. Desired power $\boldsymbol{x_t^g}$ must respect technological constraints, we simplify this constraint as: $\boldsymbol{0 \leq x_t^g \leq x_{max,t}^g}$ where $\boldsymbol{x_{max,t}^g}$ is the maximum power output that can be generated by $\boldsymbol{g}$ at time step $\boldsymbol{t}$, w.r.t. any technological constraints that are specific to the used technology (we will not go into details about technology models as it exceeds the scope of this paper). For each ESS $\boldsymbol{s}$, let $\boldsymbol{x_t^s}$ be the desired power flow for $\boldsymbol{s}$ at time step $\boldsymbol{t}$. Charge and discharge powers resp. are given by: $\boldsymbol{P_{C,t}^s = max(0, x_t^s)}$, $\boldsymbol{P_{D,t}^s = -min(0, x_t^s)}$. Power flow $\boldsymbol{x_t^s}$ is constrained by the minimum and maximum states of charge of $\boldsymbol{s}$. We simplify this constraint as $\boldsymbol{x_{min,t}^s \leq x_t^s \leq x_{max,t}^s}$. In order to overcome the problem of dependence between successive decision variables ($\boldsymbol{x_{max,t}^g}$ is function of $\boldsymbol{x_{t-1}^g}$ while $\boldsymbol{x_{min,t}^s}$ and $\boldsymbol{x_{max,t}^s}$ are function of $\boldsymbol{x_{t-1}^s}$) we choose to use decision variables $\{\boldsymbol{OM_t^s}\}_{\boldsymbol{t\in[1,T]},\ \boldsymbol{s\in S}}$ and $\{\boldsymbol{OM_t^g}\}_{\boldsymbol{t\in[1,T]},\ \boldsymbol{g\in G}}$ such as:

$$\boldsymbol{x_t^s} = \begin{cases} \boldsymbol{OM_t^s * PD_{t,max}^s \ ; \ if \ OM_t^s \geq 0} \\ \boldsymbol{-OM_t^s * PC_{t,max}^s \ ; \ else} \end{cases} \qquad \text{and} \qquad \boldsymbol{x_t^g = OM_t^g * x_{max,t}^g}$$

With $\forall\, s,t: -1 \leq OM_t^s \leq 1$ and $\forall$ g, $t: 0 \leq OM_t^g \leq 1$

3.2 Optimization problem formalization as an MDP

The real-time optimal power flows problem can be formalized as an MDP. The system whose states are observed is the MG, more specifically the controllable technologies, i.e. ESSs $\{s \in S\}$ and CGs $\{g \in G\}$.

State: At each time step t , the state is given by the concatenation of $\{x_{min,t}^s, x_{max,t}^s\}_{s\in S}$ and $\{x_{max,t}^g\}_{g\in G}$ as well as the power load P_{L_t} , the non controllable renewable generation Ren_t , the power cost Pc_t and the feed in tariff Ft_t of the UG ($€ / W$).

Action: At each time step t , an action is given by vectors $\{OM_t^s\}_{s\in S}$ and $\{OM_t^g\}_{g\in G}$

Transition: Part of the transition is deterministic (charge/discharge of ESSs and generation using CGs) and part of it is stochastic (P_{L_t}, Ren_t , Pc_t and Ft_t signals).

Reward: The reward is given by the sum of any functions that represent criteria to maximize (if the criterium needs to be minimized, we multiply the function by -1). We use normalized objective functions that include UG energy cost and energy feed-in gains, operation cost for each technology and a penalty term when the action is unfeasible (squared error between action and

violated limit). This penalty stays relatively small since we added the Sigmoid and Softsign activation functions to the output layer of the actor network.

4 PPO implementation details

Proximal Policy Optimization (PPO) is a deep actor-critic, policy gradient reinforcement learning algorithm. The main idea behind reinforcement learning (RL) is the use of a trial-and-error mechanism to learn the best policy. Deep RL means that the algorithm includes the use of one or many neural networks. Policy gradient means that the actor (which is a neural network in this case) undergoes incremental updates using a chosen performance evaluation function that needs to be optimized (which is why neural networks come in handy since this optimization is done using gradient ascent). Finally, actor-critic deep RL algorithms use two neural networks: the actor network selects actions for every state, the critic receives the trajectories built using the actor and estimates the value function for each < state, action > tuple. The value function is used to compute feed-back that is sent to the actor to learn from. This back-and-forth process allows the optimization of the cumulative reward. Detailed algorithm is given in [1].

4.1 Actor and critic networks

In our implementation of PPO for our specific problem (power flow optimization in the MG) we used the networks shown in figures 4.1 and 4.2. We started by implementing the actor and critic networks presented in [5] then added normalization layers to apply the observation normalization recommendation in [2]. Input layers, hidden layers and output layers are linear. We used the Pytorch Python library to implement these models. The choice of activation functions for the actor's output layer is based on the desired sign of the action. Controllable generators receive an action between 0 and 1 (hence the Sigmoid activation) while storage technologies receive an action between -1 and 1 (hence the Softsign activation). L_a and L_s are the action vector and state vector lengths resp. and S_{L_s-1}, S_{L_s} are the power cost Pc_t and the feed in tariff Ft_t of the UG resp.

4.2 Algorithm optimizations

In the following, we try to visualize the impact of each algorithm optimization on the convergence of the model and the final cumulative reward.

4.2.1. Orthogonal initialization

We use orthogonal initialization for the layers of the actor and critic networks. The objective is to solve the vanishing and exploding gradient problems.

4.2.2. Observation normalization

The state isn't fed directly into the input layer, we instead normalize it using a normalization layer. We normalize quantities that are semantically homogenous, i.e. quantities that represent the power load, minimum and maximum power flows for every controllable technology, and uncontrollable renewable power.

4.2.3. Adam learning rate annealing

Some tasks in RL literature benefit from learning rate annealing in terms of cutting down the training time and improving performance. We test it for our problem.

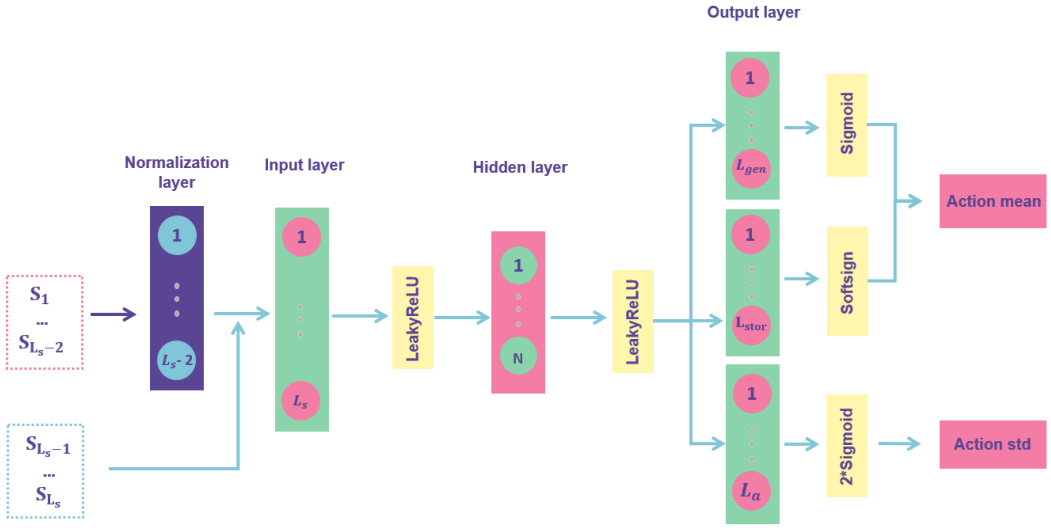

Figure 4.1: Actor network

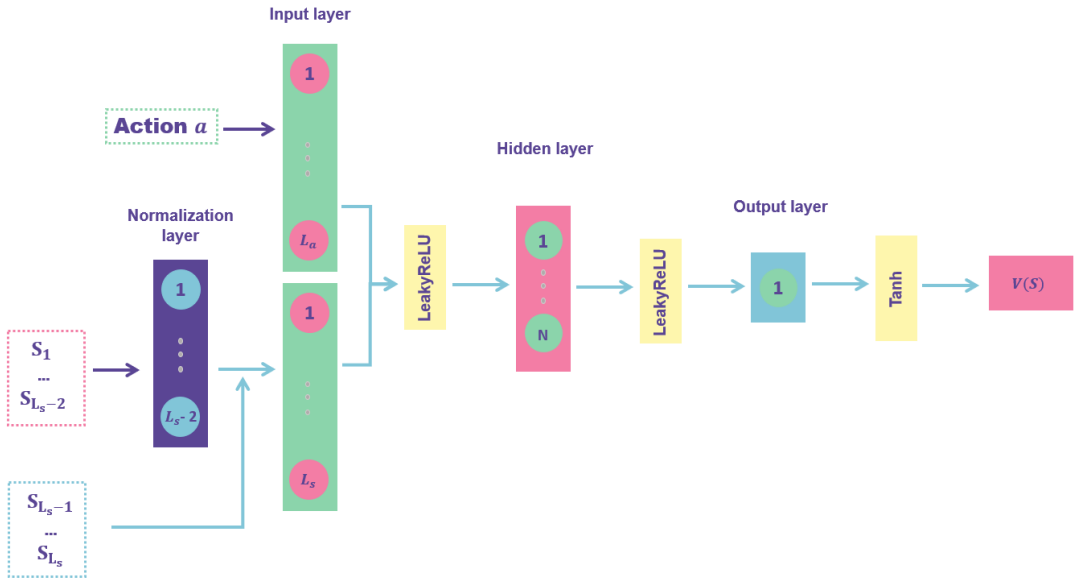

Figure 4.2: Critic network

5 Results

Figure 5.1 gives the visualization of a test run on the trained RL agent. Power loads, uncontrollable

renewable generation, energy costs and feed-in tariffs of the UG are received real-time by the agent. The MG is composed of two batteries and three controllable generators (two biomass and one gas generator). Interpretation of the optimal power flows w.r.t. chosen objective functions and their scaling is important in order to analyze the performance of the algorithm but we will not discuss this aspect as it is not the objective of the article. Figures 5.2, 5.3, 5.4 and 5.5 represent the evolution of the cumulative reward for each combination of algorithm optimizations: orthogonal initialization of layers (1), observation normalization (2) and Adam learning rate annealing (3). From fig. 5.5 we can see that the most impactful optimization is the normalization of observations, since there's no convergence without it. From fig. 5.2 and 5.3 we can see that learning rate annealing allows faster convergence. From fig. 5.4 we can see that removing orthogonal initialization improves cumulative reward, but slows down convergence.

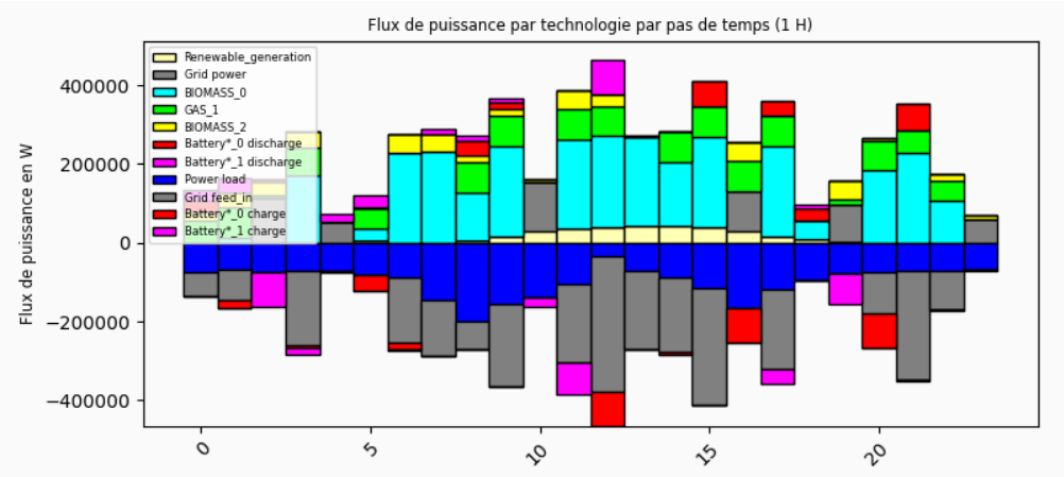

Figure 5.1: Real-time power flows on test run. Renewable generation is under-sized compared to power loads. Power flows are balanced (total +/- flows are equal)

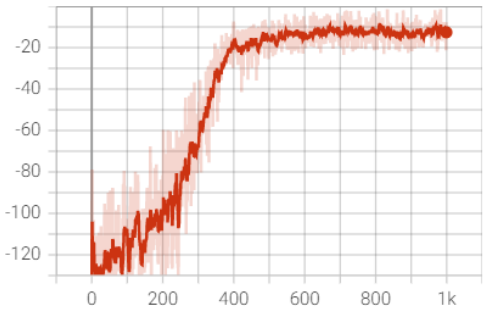

Figure 5.2: Evolution of cumulative reward with optimizations 1 and 2

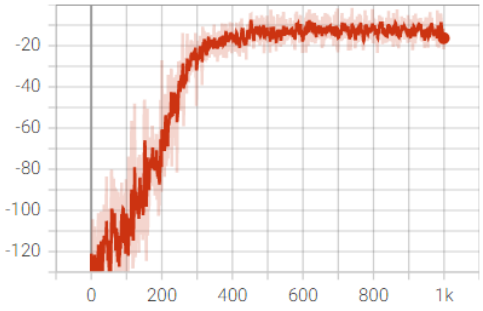

Figure 5.3: Evolution of cumulative reward with optimizations 1, 2 and 3

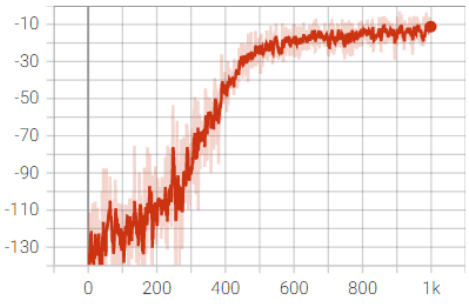

Figure 5.4: Evolution of cumulative reward with optimizations 2 and 3

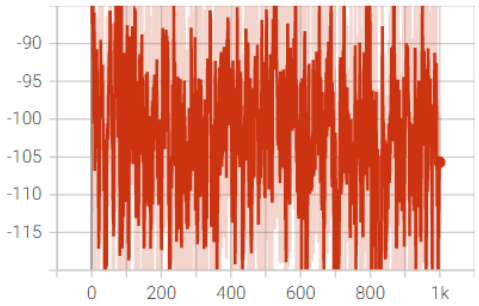

Figure 5.5: Evolution of cumulative reward with optimizations 1 and 3

6 Challenges and future directions

The main challenge in solving the real-time optimal power flows problem has been the choice of action vector and the sizing of the actor and critic networks to fit our specific environment (the MG). Actions in [5] represent power flows. However, unlike the authors, we were not able to achieve model convergence within 500 learning iterations since the actor was not able to generate feasible actions w.r.t. technological constraints. There can be many explanations for this, as the article didn't discuss implementation details such as the choice of layers and activations and why they're adapted to their problem formalization. This is a pretty common challenge, which is why we tried to give as much details and directions to reimplement our work. Regarding future directions, we would like to emphasize the importance of explainability of the optimization results. Explainability in deep RL has been gaining a lot of interest lately, specifically Post-Hoc explainability, i.e. analyzing results after the deep RL algorithm finishes training [6]. We believe that trustworthiness of optimization algorithms is most important in MG control. Stakeholders need to be able to understand the causality between the inputs (e.g., demand, meteorological phenomena, energy price signals, etc.) and outputs (MG control decisions) in an intuitive manner, as opposed to trusting an algorithm solely because of the promises of the scientific community, since obviously, the stakes are higher than in other deep RL application fields like training an agent to play board games. We therefore believe that our future work should be directed towards this matter.

References

[1] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov, "Proximal Policy Optimization Algorithms", doi: https://doi.org/10.48550/arXiv.1707.06347

[2] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, Aleksander Madry, "Implementation Matters in Deep Policy Gradients: A Case Study on PPO and TRPO", doi: https://doi.org/10.48550/arXiv.2005.12729

[3] Kepa Solaun, Emilio Cerdá, "Climate change impacts on renewable energy generation. A review of quantitative projections", Renewable and Sustainable Energy Reviews, Volume 116, 2019, doi: https://doi.org/10.1016/j.rser.2019.109415

[4] Ministère de la Transition écologique et de la Cohésion des territoires, Ministère de la Transition énergétique, « Rapport : Travaux relatifs au nouveau nucléaire », février 2022, URL : https://www.ecologie.gouv.fr/sites/default/files/2022.02.18_Rapport_nucleaire.pdf

[5] Chenyu Guo, Xin Wang, Yihui Zheng, Feng Zhang, "Real-time optimal energy management of microgrid with uncertainties based on deep reinforcement learning", Energy, Volume 238, Part C, 2022, doi: https://doi.org/10.1016/j.energy.2021.121873

[6] Alexandre Heuillet, Fabien Couthouis, Natalia Díaz-Rodríguez, "Explainability in Deep Reinforcement Learning", doi: https://doi.org/10.48550/arXiv.2008.06693