A Reinforcement Learning Framework for Optimisation of Power Grid Operations and Maintenance

R. Rocchetta ,[1*] M. Compare ,[1,2] E. Patelli ,[1] and E. Zio [2,3,4,5]

[1] *Institute for Risk and Uncertainty, Liverpool University, United Kingdom*
[2] *Aramis s.r.l., Milano, Italy*
[3] *Energy Department, Politecnico di Milano,Italy*
[4] *Chair on Systems Science and the Energetic challenge,Foundation Electricite' de France at CentraleSupelec, France*
[5] *Eminent Scholar, Department of Nuclear Engineering, College of Engineering, Kyung Hee University, Republic of Korea*
[*] Corresponding author: roberto.rocchetta@liverpool.ac.uk

Abstract

In this work, Reinforcement Learning (RL) is used for managing operation and maintenance of power grids equipped with Prognostic and Health Management (PHM) capabilities, which allow tracking the health state of the grid components. RL exploits this information to select optimal state-action-reward trajectories maximising the expected profit by selecting proper operation and maintenance actions on the grid components. A scaled down cost study is solved for a power grid, and strengths and weaknesses of the framework discussed.

Keywords: Reinforcement Learning, Prognostic and Health Management, Markov Decision Process, Degradation, Power Grid, Uncertainty

1 Introduction

Modern power grids are complex systems, including many highly interconnected components. Maximising the grid productivity while assuring a safe and reliable delivery of power is of uttermost importance for grid operators. This requires developing robust decision-making frameworks, which give account to both the complexity of the asset and the uncertainties on its operational conditions, component degradation and failure behaviors, external environment, etc.

Nowadays, the grid management issue is further challenged by the possibility of equipping grid elements with Prognostics and Health Management (PHM) capabilities, which allow tracking the health state evolution. This information can be exploited by grid operators to further increase the profitability of their assets [1–6].

Reinforcement Learning (RL) [7, 8] has been used in the last decades to solve a variety of realistic control and decision-making issues in the presence of uncertainty, including power grid management. In the RL paradigm, a controller (i.e. the decision maker) interacts with the environment (e.g. the grid) by observing states, collecting rewards and selecting actions to maximise the future reward. The state-action-reward trajectories [9] can be gathered from direct interaction with a real system (e.g. [10]), or from a realistic simulator of the environment [7], also encoding the aleatory uncertainties in the system future behavior. This makes RL suitable to power grid management optimization, as it can cope with both the complexity of the asset and the unavoidable uncertainties related to its operation.

In [6], an RL framework based on Q-learning is proposed to solve constrained load flow and reactive power control problems in power grids. Kuznetsova et al. [5] develop an optimisation scheme for consumers actions management in the microgrid contest and accounting for renewable volatility and environmental uncertainty. In [9], a comparison between RL and a predictive control model is presented for a power grid damping problem. In [4], the authors review recent advancements in intelligent control of micro grids including few attempts using RL methods. However, none of the revised works employs RL to find optimal combined Operation and Maintenance (O&M) policies for power grids with degrading elements.

We present an RL framework to support O&M decisions for power grids equipped with PHM systems, which seeks for the settings of the generator power outputs and the scheduling of preventive maintenance actions that maximize the grid load balance and expected profit over an infinite time horizon, while considering the uncertainty of renewable energy sources, power loads and component failure behaviors.

The rest of this paper is organized as follows: Section 2 presents the RL framework for optimal decision making under uncertainty is described. A scaled-down power grid application is proposed in Section 3 and results of the MDP and RL algorithm compared and discussed in Sections 4 and 5. Section 6 closes the paper.

2 Modelling framework for optimal decision making under uncertainty

As anticipated above, developing a RL framework for power grid O&M management requires defining the environment, the actions that the agent can take in every state of the environment, the state transitions the actions lead to and, finally, the rewards associated to each state-action-transition step.

2.1 State space

Consider a power grid made up of elements $C = \{1, ..., N\}$ physically and/or functionally interconnected, according to the given grid structure. Similarly to [13], the features of the grid elements defining the environment are the N^d degradation mechanisms affecting the degrading components $d \in D \subseteq C$ and the N^p possible setting variables of power sources $p \in P \subseteq C$. For simplicity, we assume $D = \{1, ..., |D|\}$ and $P = \{|D| + 1, ..., |D| + |P|\}$.
The degradation processes evolve independently on each other according to a Markov process defining the transition probability from state $s_i^d(t)$ at time t to the next state $s_i^d(t+1)$, where $s_i^d(t) \in \{1, ..., S_i^d\} \quad \forall t$, $d \in D, i = 1, ..., N^d$. Similarly, for the power sources production, a Markov process defines the probabilistic dynamic from $s_j^p(t)$ at time t to the next state $s_j^p(t+1)$, where $s_j^p(t) \in \{1, ..., S_j^p\} \quad \forall t, p \in P, j = 1, ..., N^p$. Then, state vector $\mathbf{S}(t)$ at time t reads:

$$\mathbf{S}(t) = \left[\; s_1^1(t), s_2^1(t) \ldots s_{N^{|P|+|D|}}^{|P|+|D|}(t) \; \right] \tag{1}$$

2.2 Actions

Actions can be performed on the grid components at each t, which define the system action vector as follows:

$$\mathbf{a}(t) = \left[\; a_1(t) \ldots a_c(t) \ldots a_{N_c}(t) \; \right] \tag{2}$$

were action $a_c(t)$ is selected for component $c \in D \cup P$ among a set of mutually exclusive actions $a_c \in \mathbf{A}_c$. The action set $\mathbf{A}_c$ can include operational actions (e.g. closure of a valve, generator power ramp up, etc.) and maintenance actions (e.g. preventive and corrective). Constraints can be defined for reducing $\mathbf{A}_c$ to subset $\mathbf{A}_c(s_f^c) \subseteq \mathbf{A}_c$. For example, Corrective Maintenance (CM), cannot be taken on As-Good-As-New (AGAN) components and, similarly, it is mandatory action for failed components. In an optimistic view [13], both Preventive Maintenance (PM) and CM actions are assumed to restore the AGAN state for each component. An example of Markov process for a 4 degradation state component is presented in Fig.1, where circle markers indicate maintenance actions and squared markers indicate other actions, i.e. operational actions.

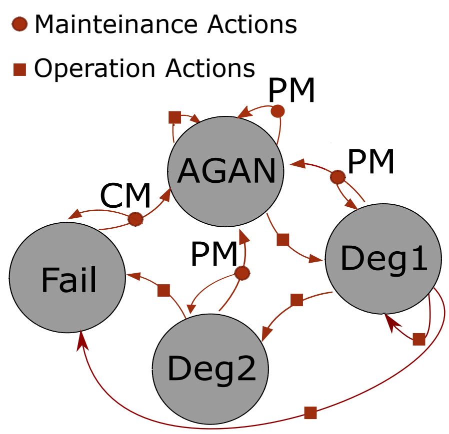

Figure 1: The Markov Decision Process associated to the health state of a degrading component.

2.3 Transition probabilities

Transition probability matrices are associated to each component feature and action as follows:

$$\mathcal{P}_{c,f,a} = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,1} & p_{n,2} & \cdots & p_{n,n} \end{bmatrix}_{c,f,a} \tag{3}$$

where $p_{i,j}$ represents the probability of transition from state i to state j of feature f of component c and conditional to the action a in a time varying setting $\mathcal{P}_{c,f}(s_j|a,s_i)$. The normalization propriety holds, i.e. $\sum_{j=1}^{n} p_{i,j} = 1$. In practice, element $p_{i,j}$ of the transition probability matrix $\mathcal{P}_{c,f,a}$ can be estimated as the relative frequency of the measured component state to fall into the j^{th} state at time $t+1$ provided that it was at the i^{th} state in the previous time step when the action a was taken.

2.4 Rewards

Numerical rewards are case-specific and obtained by solving a physic-economic model of the system, which evaluates how good is the transition from one state to another given that a is taken:

$$R = \mathfrak{F}\left(\mathbf{S}(t+1)\ ,\ \mathbf{a}(t)\ ,\ \mathbf{S}(t)\right)\ \ \in \mathbb{R}$$

2.5 Reinforcement Learning and SARSA(λ) method

Generally speaking, the goal of RL methods for optimal control is to find the optimal action-value function $Q_{\pi^*}(s,a)$, which is an indicator of future revenues when an action a is taken in state s, following the optimal policy π^*:

$$Q_{\pi^*}(\mathbf{S},A) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} R_t|\mathbf{S}(t),A(t)\right] \tag{4}$$

Among the wide range of RL algorithms, we adopt SARSA(λ) which is a temporal difference learning methods (i.e. it changes an earlier estimate of Q based on how it differs from a later estimate) employing eligibility traces to carry out backups over n-steps and not just over one step [7]. Details on SARSA(λ) are provided in the Appendix.

3 Case study

A scaled-down power grid case study is used to test the decision making framework. The grid includes 4 nodes and 5 cables for the power transmission, 2 non-controllable Renewable Energy Sources (RES) are connected to 2 loads (nodes 2 and 3) and provide them electric power depending on random weather conditions (Fig. 2). Two traditional generators installed at nodes 1 and 4 are controlled to minimize power unbalances on the grid. We assume that the 2 controllable generators and links 1-2 and 1-3 are affected by degradation and, thus, are equipped with PHM capabilities to inform the decision-maker on their degradation states.

3.1 States and Actions

In the considered case study, we associate features to $N_c = 8$ components: the 2 loads, the 2 renewable generators, 2 transmission lines and the 2 controllable generators, each of them is associated to only 1 feature ($N_{fc} = 1$ or $N_O = 1$). For each load, we consider $N_{fsc} = 3$ states which are identified by index $I_L \in \{$ low, medium, high $\}$ power demand. Three states are associated to renewable power, identified by the index $I_{RES} \in \{low, medium, high\}$ production, and 3 health states are associated to the transmission links $H_l \in \{AGAN, degraded, failed\}$. Finally, 4 degradation states H_G are considered for the 2 generators in nodes 1 and 4 $H_G \in \{AGAN, degraded, highly\ degraded, failed\}$. Then, the total number of state vectors combinations is 11664. An example of system state vector at time t for is as follows:

$$\mathbf{S}(t) = \{H_{G,1}, H_{G,4}, I_{L,2}, I_{L,3}, I_{RES,2}, I_{RES,3}, H_{l,1-2}, H_{l,1-3}\}_t$$

$$H_G \in \{1,2,3,4\};\ \ H_l \in \{1,2,3\}$$

$$I_L \in \{1,2,3\};\ \ I_{RES} \in \{1,2,3\}$$

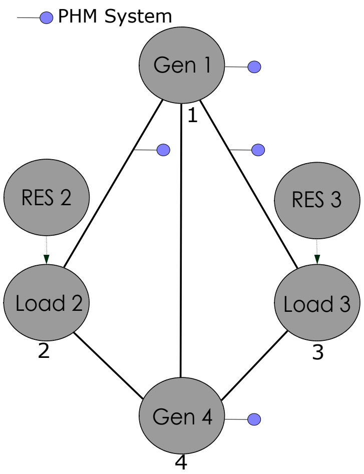

Figure 2: The power grid structure and the position of the 4 PHM capabilities, 2 renewable sources, 2 loads and 2 traditional generators.

where H_G, H_l, I_L and I_{RES} are the degradation state (health indicator) of the generators and cables and the indices identifying the load demand and renewable power production states, respectively.

The agent can operate both generators with the aim to maximise the system revenue by minimizing unbalance between demand and production, while preserving the structural and functional integrity of the system. Other actions can be performed by other agents on other components (e.g. transmission lines), but being outside from the control domain of the first agent those are assumed included in the environment. Five actions can be performed on each controllable generator, for a total of 25 combinations, thus giving rise to a 291600 state-action pairs. The action set for each generator is the following:

$$\mathbf{A}_g = \{go2P_{g1}, go2P_{g2}, go2P_{g3}, PM, CM\}$$

where the first 3 (operational) actions affect the power output of the generator, changing it to one of the 3 allowed power levels. The last 2 actions are preventive and corrective maintenance actions. It is assumed that CM has to be taken only for failed generators. Conversely, PM can be performed if the generator is degraded but not when it is failed or AGAN. Furthermore, highly degraded generators (i.e. $H_G = 3$) are assumed degraded in their operational performance and only the lower power output can be obtained (only $go2P_{g1}$ is allowed). Tables 1-3 display the costs for each action and the corresponding power output of the generator, the line electric parameters and the relation between state indices and physical values for the RES and loads, respectively.

Table 1: The power output of the 2 generators in [MW] associated to the 5 available actions and action costs in monetary unit [m.u.].

Action:	$go2P_{g1}$	$go2P_{g2}$	$go2P_{g3}$	PM	CM
$P_{g,1}$ [MW]	40	50	100	0	0
$P_{g,4}$ [MW]	50	60	120	0	0
$C_{a,g}$ [m.u.]	0	0	0	10	500

Table 2: The transmission lines proprieties.

From	To	Ampacity [A]	Reactance
1	2	125	0.0845
1	3	135	0.0719
1	4	135	0.0507
2	4	115	0.2260
3	4	115	0.2260

Table 3: The generators outputs in MW and costs in monetary unit (m.u.) associated to the 5 available actions.

State Index	1	2	3
$P_{RES,2}$ [MW]	0	20	30
$P_{RES,3}$ [MW]	0	20	60
$P_{L,2}$ [MW]	60	100	140
$P_{L,3}$ [MW]	20	50	110

3.2 Probabilistic Model

State transitions may occur from time t to the next time step $t+1$ and are specifically defined for each feature of each component. The 2 loads have identical transition probability matrices and also the degradation of the transmission cables and generators is described by the same Markov process. Thus, for ease of notation the node subscripts have been dropped. Each action $a \in \mathbf{A}_g$ is associated to a specific transition probability matrix $\mathcal{P}_{g,a}$ describing the evolution of the generator health state conditioned by its operative state or maintenance action. It can be noticed that probabilities associated to operational actions, namely $go2P_{g1}$, $go2P_{g2}$, $go2P_{g3}$, affect differently the degradation of the component. For those actions, the bottom row corresponding to the failed state has only zero entries. This reflects the fact that operational actions cannot be taken for failed generators, but only CM is allowed. The transition matrices for the 8 considered states are defined as follows:

$$\mathcal{P}_{g,go2P_{g1}} = \begin{pmatrix} 0.98 & 0.02 & 0 & 0 \\ 0 & 0.95 & 0.05 & 0 \\ 0 & 0 & 0.9 & 0.1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad \mathcal{P}_{g,go2P_{g2}} = \begin{pmatrix} 0.97 & 0.03 & 0 & 0 \\ 0 & 0.95 & 0.05 & 0 \\ 0 & 0 & 0.9 & 0.1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathcal{P}_{g,go2P_{g3}} = \begin{pmatrix} 0.95 & 0.04 & 0.01 & 0 \\ 0 & 0.95 & 0.04 & 0.01 \\ 0 & 0 & 0.97 & 0.03 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad \mathcal{P}_{g,PM} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 \\ 0.5 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathcal{P}_{g,CM} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.15 & 0 & 0 & 0.85 \end{pmatrix}$$

$$\mathcal{P}_{RES,2} = \begin{pmatrix} 0.5 & 0.1 & 0.4 \\ 0.3 & 0.3 & 0.4 \\ 0.1 & 0.4 & 0.5 \end{pmatrix} \qquad \mathcal{P}_{RES,3} = \begin{pmatrix} 0.5 & 0.2 & 0.3 \\ 0.4 & 0.4 & 0.2 \\ 0 & 0.5 & 0.5 \end{pmatrix}$$

$$\mathcal{P}_{Load} = \begin{pmatrix} 0.4 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.4 \\ 0.2 & 0.4 & 0.4 \end{pmatrix} \qquad \mathcal{P}_{lines} = \begin{pmatrix} 0.9 & 0.08 & 0.02 \\ 0 & 0.97 & 0.03 \\ 0.1 & 0 & 0.9 \end{pmatrix}$$

3.3 Reward Model

When the agent performs an action at time t; the environment provides a reward and leads the system to its state at time $t+1$. The reward is calculated as a sum of 4 different terms: (1) the cost of not serving energy to the customers, (2) the revenue from selling electric power, (3) the cost of producing electric power with traditional generators and (4) the cost associated to the performed actions. Mathematically, the reward reads:

$$R(t) = +\sum_i (L_i - ENS_i/\Delta_t) \cdot C_{el} + \sum_g P_g \cdot C_g - \sum_g C_{a,g} - \sum_i ENS_i \cdot C_{ENS} \tag{5}$$

where ENS_i is the energy not supplied to the node i (computed uby DC power flow [14]), C_{ENS} is the cost of the energy not supplied, L_i is the power demanded by node i, C_{el} is the price paid by the loads for per-unit of electric power, P_g is the power produced by the generators, C_g is the cost of producing the unit of power, $C_{a,g}$ is the cost of the action a on the generator g and Δ_t is the time difference between the present and the next system state and it is assumed to be 1 h. The costs C_{ENS}, C_g and C_{el} are set to 5, 4 and 0.145 monetary unit (m.u.) per-unit of energy or power, respectively.

4 Results and Discussions

4.1 MDP

The Bellman's optimality equation has been solved using the value-iteration dynamic programming method [7], which gives the optimal action-value function $Q_{\pi^*}(s,a)$. This is summarise in Fig. 3, where the curves provide a compact visualization of the distribution of $Q_{\pi^*}(s,a)$ over the states for the available 25 combinations of actions. Three clusters can be identified: on the far left, we find the set of states from which CM on both generators is performed; being CM a costly action, this leads to a negative expectation of the discounted reward. The second cluster (*C 2*) corresponds to the 8 combination of one CM and any other action on the not failed generator. The final cluster (*C 1*) of 16 combinations of actions includes only PM and operational actions. If corrective maintenance is not performed, higher rewards are expected. Indeed, this is not always possible as generators can randomly transit to a failed state.

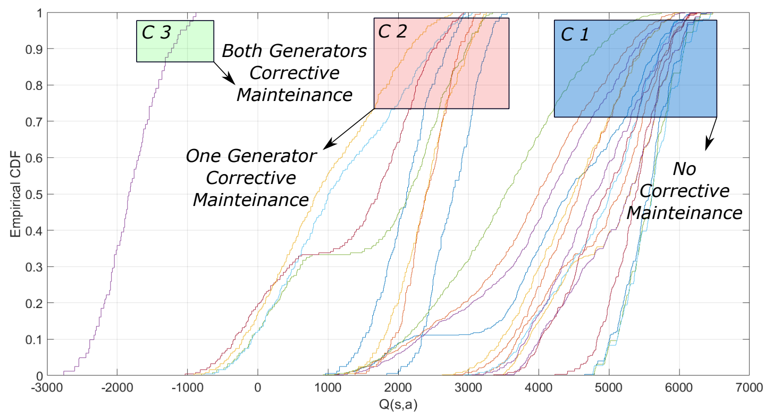

Figure 3: The $Q(s,a)$ values displayed using ECDFs and the 3 clusters.

In Fig. 4 each sub-plot shows the the highest $Q_{\pi^*}(s,a)$ expected discounted power grid return adopting the optimal policy, conditional to a specific degradation states of the generators and for increasing electric load demand. It can be noticed that if the generators are both healthy or slightly degraded (i.e. $H_{G,1} + H_{G,2}$ equal 2, 3 or 4) an increment in the overall load demand leads to an increment in the expected reward, due to the larger revenues from selling more electric energy to the customers. On the other hand, if the generators are highly degraded or failed (i.e. $H_{G,1} + H_{G,2}$ equal 7 or 8), an increment in the load demand leads to a drop in the expected revenue. This is due to the increasing risk of load curtailments and associated costs, i.e. cost of energy not supplied, and to impacting PM and CM actions costs.

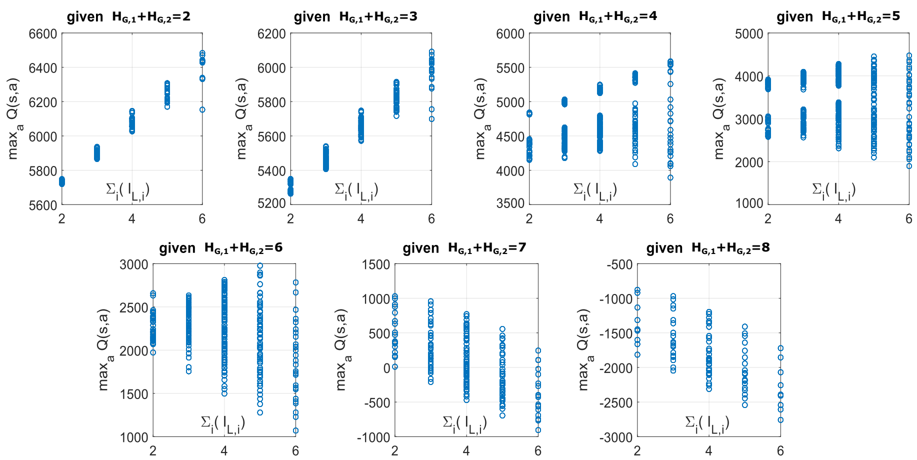

Figure 4: The maximum $Q(s,a)$ (i.e. maximum expected discounted cumulative reward) for increasing total load and different degrading condition of the generators.

4.2 SARSA(λ)

The SARSA(λ) algorithm (Algorithm 1 in the Appendix) has been used to provide an approximate solution to the MDP associated with the decision problem. The MDP is used to sample control trajectories only, i.e. it provides a reward and a new state when an action and the old state is provided as input. The SARSA method has been run changing parameters setting and accumulating eligibility traces. According to the SARSA algorithm, an initial state has to be selected for the episodic loop, e.g. randomly. In this

Table 4: The MDP Bellman's optimality and the RL results compared with suboptimal policies.

	MDP	SARSA(0.5)		Q_{50rnd}	Q_{100rnd}
Q_{s1} [m.u.]	5719	5511	5555	4191	2028
Q_{s2} [m.u.]	2898	2577	2664	1297	-1229
Q_{s3} [m.u.]	-1721	-1816	-1813	-2956	-4288
Act top1	100 %	48.8 %	49.1 %	62.1%	24.8%
Act top3	100 %	66.5 %	66.5 %	71.4%	43.1%
$\mathbb{E}[R(t)]$ [m.u.]	529.8	478.8	488.1	370.3	190.4
N_e	-	5e5	5e5	-	-
T	-	50	250	-	-

work, the initial system state $\mathbf{S}(t=0)$ is sampled form S using a degradation-weighted probability mass function $f_S(s) = \sum_{c=1}^{N_c}(H_{c,s})/\sum_{s=1}^{N_s}[\sum_{c=1}^{N_c}(H_{c,s})]$. This sampling scheme is used to better estimate action-value functions in rarely visited sates (i.e. low-probability states with many failed/highly degraded components) and thus speed up the convergence of the SARSA method.Three representative states system states $s_1 = [1,1,1,1,1,1,1,1]$, $s_2 = [4,1,1,1,1,1,1,1]$ and $s_3 = [4,4,3,3,3,3,3,3]$ are analysed and compared with the MDP reference solution. The 3 states have substantially different expected discounted rewards, s_1 has both generators in a AGAN state, s_2 has on generator out of service whilst s_3 has both generators failed and have been selected from the 3 clusters *C 1*, *C 2* and *C 3*, respectively (see Fig. 3).

4.3 Policies comparison

Table 4 shows the results for the MDP (Bellman's optimality) and compares it with SARSA runs. *Act* is defined as the portion of actions taken from the SARSA policy that are equal those taken using the reference MDP optimal policy in the corresponding states; $\mathbb{E}[R(t)]$ is the expected non-discounted return, independent from the initial state of the system. Trial and error testing showed that SARSA(0) policies were outperformed by SARSA(0.5) results and, thus, two SARSA(λ) with $\lambda = 0.5$ have been further investigated by setting the truncation windows T for each episode to 50 and 250 time steps, respectively. A suboptimal policy Q_{50rnd} was artificially obtained randomizing the action to be selected in 50 % of the states. For Q_{100rnd} all states have a random action associated with. It is interesting to notice that SARSA(0.5) provide better policies (i.e. higher expected discounted and non-discounted returns) compared to Q_{50rnd} and Q_{100rnd}. This is true even if Q_{50rnd} has higher *Act* compared to the SARSA policies, i.e. more than 60 % of the Q_{50rnd} actions are equal to the MDP actions whilst less than 50 % for the SARSA. This points out that the optimal policy is very sensitive to some of the state-action combinations and less to others. In other words, taking the wrong action in some states can lead to a catastrophic drop in the expected return, whilst in other cases a sub-optimal action affects less the expected revenue (e.g. making generator 1 produce power rather than generator 2 or vice versa).

Fig. 5 presents in detail 2 control trajectories with associated rewards and actions. MDP and SARSA policies were used greedily to select actions and the results are displayed in the top and bottom figures, respectively. Table 5 display the corresponding 10 vectors of grid state indices used to select the actions. It is interesting to observe that by following the MDP policy, two PM actions were selected, first on generator 1 (at the time step 2) and then on the generator installed on the node 4 (at the time step 3). The PM actions have been recommended even if the generators were in a AGAN state. This might seem counter intuitive, but it can be explained considering the degradation model settings. A preventive maintenance action taken in an AGAN degradation state will assure a transition to the AGAN state. In this sense, the MDP policy is sometimes ready to accept a slightly lower revenue (due to PM costs) but with the advantage of suspending the degradation process. Also, no additional costs are due to unbalances between power load and production in those scenarios, even if one of the generators is unavailable due to PM. This is due to the higher production from renewable sources in node 3 (at time step 2) and both RES in nodes 2 and 3 (at time step 3). Similar argument is also valid for the SARSA control trajectory at the time step 7.

5 Discussion on Limitation

While RL, like stochastic dynamic programming (DP), has in principle a very broad scope of application, it has to face similar issues when the state-action spaces of the control problem are very large. In such

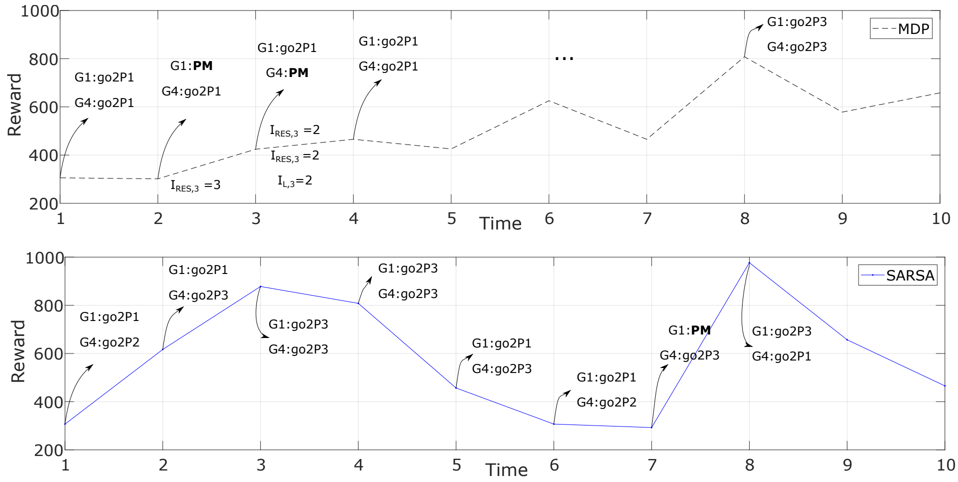

Figure 5: Actions taken in 2 separate control trajectories using MDP and SARSA policies. Initial state s_1 and next states are randomly generated by the underlying probabilistic model (see Table 5).

Table 5: The state indices for the MDP control trajectory in Figure 5.

MDP							
$H_{G,1}$	$H_{G,4}$	$I_{L,2}$	$I_{L,3}$	$I_{RES,2}$	$I_{RES,3}$	$H_{l,1-2}$	$H_{l,1-3}$
1	1	1	1	1	1	1	1
1	1	1	1	3	1	1	1
1	1	1	2	2	3	1	1
1	1	2	1	3	2	1	1
1	1	1	2	3	1	1	1
1	1	3	1	3	3	3	1
2	1	2	1	3	2	1	1
2	2	2	3	1	2	1	1
2	2	2	2	1	1	1	1
2	2	1	3	1	2	1	1

a case, RL has to be combined with regression techniques capable of interpolating over the state-action space the data obtained from (relatively) few control trajectories [9]. Most of the research in this context has focused on parametric function approximators, representing either some (state-action) value functions or parameterized policies, together with some stochastic gradient descent algorithms (see e.g. [8] or [15]). In real world environments, it is highly unlikely to have a complete description of the system state and the Markov property just rarely holds. Partially observability is formally defined in Markov Decision Processes to better describe the dynamics of system by acknowledging a lack of information of some of the states in the system. As consequence, the issue of partial observability inevitably affects several RL applications and just few works attempted to tackle the problem, e.g. [16]-[17]-[18]. To conclude, further research has to be devoted to the development of enhanced RL algorithms, capable of dealing with imprecise rewards (e.g. due to unavailable/unreliable models), partial observability and issues related to scarcity of samples due to low-probability of specific state-action pairs.

6 Conclusion

A framework for optimal decision making of power grid systems affected by uncertain operations and degradation mechanisms has been presented. The framework is based on Markov decision process theory and Reinforcement Learning algorithms. Power grid models can include prognostic health management devices which are used to inform the agent about the health state of the system components. This information helps to select between operational and maintenance actions which have to be taken on the system components. The SARSA(λ) method was used to solve a control problem for a scale down power grid with renewable and PHM capabilities. The reinforcement learning results have been compared to the reference Bellman's optimality solution and are in good agreement, although inevitable approximation errors have been observed. The framework proved to be flexible and effective in tackling a small but representative case study and future works will test its applicability to more realistic (larger) state-action spaces. For this aim, artificial neural networks can be used for state-action space regression and this will hopefully allow to scale up to larger grids. This necessary verification for a possible future applicability of the method.

Acknowledgments

The authors would like to acknowledge the gracious support of this work through the EPSRC and ESRC Centre for Doctoral Training on Quantification and Management of Risk & Uncertainty in Complex Systems & Environments Grant number (EP/L015927/1)

References

[1] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, Review of road traffic control strategies, *Proceedings of the IEEE* **91**, 2043 (2003), ISSN 0018-9219.

[2] J. Jin and X. Ma, Hierarchical multi-agent control of traffic lights based on collective learning, *Engineering Applications of Artificial Intelligence* **68**, 236 (2018), ISSN 0952-1976.

[3] R. Yousefian, R. Bhattarai, and S. Kamalasadan, Transient stability enhancement of power grid with integrated wide area control of wind farms and synchronous generators, *IEEE Transactions on Power Systems* **32**, 4818 (2017), ISSN 0885-8950.

[4] M. S. Mahmoud, N. M. Alyazidi, and M. I. Abouheaf, Adaptive intelligent techniques for microgrid control systems: A survey, *International Journal of Electrical Power & Energy Systems* **90**, 292 (2017), ISSN 0142-0615.

[5] E. Kuznetsova, Y.-F. Li, C. Ruiz, E. Zio, G. Ault, and K. Bell, Reinforcement learning for microgrid energy management, *Energy* **59**, 133 (2013), ISSN 0360-5442.

[6] J. G. Vlachogiannis and N. D. Hatziargyriou, Reinforcement learning for reactive power control, *IEEE Transactions on Power Systems* **19**, 1317 (2004), ISSN 0885-8950.

[7] R. S. Sutton, D. Precup, and S. Singh, Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning, *Artificial Intelligence* **112**, 181 (1999), ISSN 0004-3702.

[8] C. Szepesvari, *Algorithms for Reinforcement Learning* (Morgan and Claypool Publishers, 2010), ISBN 1608454924, 9781608454921.

[9] D. Ernst, M. Glavic, F. Capitanescu, and L. Wehenkel, Reinforcement learning versus model predictive control: A comparison on a power system problem, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **39**, 517 (2009), ISSN 1083-4419.

[10] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming* (Athena Scientific, 1996), 1st ed., ISBN 1886529108.

[11] E. Gross, On the bellmans principle of optimality, *Physica A: Statistical Mechanics and its Applications* **462**, 217 (2016), ISSN 0378-4371.

[12] R. S. Sutton and A. G. Barto, Reinforcement learning i: Introduction (2017).

[13] M. Compare, P. Marelli, P. Baraldi, and E. Zio, A markov decision process framework for optimal operation of monitored multi-state systems, *Proceedings of the Institution of Mechanical Engineers Part O Journal of Risk and Reliability* (2018).

[14] R. Rocchetta and E. Patelli, Assessment of power grid vulnerabilities accounting for stochastic loads and model imprecision, *International Journal of Electrical Power & Energy Systems* **98**, 219 (2018), ISSN 0142-0615.

[15] H. Li, T. Wei, A. Ren, Q. Zhu, and Y. Wang, in *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)* (2017), 847–854.

[16] S. D. Whitehead and L.-J. Lin, Reinforcement learning of non-markov decision processes, *Artificial Intelligence* **73**, 271 (1995), ISSN 0004-3702, computational Research on Interaction and Agency, Part 2.

[17] F. Doshi-Velez, J. Pineau, and N. Roy, Reinforcement learning with limited reinforcement: Using bayes risk for active learning in pomdps, *Artificial Intelligence* **187-188**, 115 (2012), ISSN 0004-3702.

[18] R. Lieck and M. Toussaint, Temporally extended features in model-based reinforcement learning with partial observability, *Neurocomputing* **192**, 49 (2016), ISSN 0925-2312, advances in artificial neural networks, machine learning and computational intelligence.

[19] S. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvári, Convergence results for single-step on-policy reinforcement-learning algorithms, *Machine Learning* **38**, 287 (2000), ISSN 1573-0565.

Appendix

The SARSA(λ) algorithm starts initializing the action-value function Q and eligibility traces Z tables. Then, the values for the learning rate α, the discount factor γ, the decay rate of the traces $\lambda \in [0, 1]$ and the greediness factor ϵ (or a policy π to be evaluated) are selected. After this initialization, the episodic loop starts with a random sample (or selection) of an initial state s_t, then, an action a_t is selected based on the adopted policy, e.g. ϵ-greedy or $\pi(\cdot|s_t)$. A ϵ-greedy policy consists of random actions, taken with probability ϵ, or greedy actions taken with probability 1-ϵ (i.e. actions for which Q is maximised). Once the initial state-action pair is obtained, the episode e is evaluated (i.e. a sequence of action-rewards-state-actions). Temporal difference errors δ_t at the time step t are calculated, traces replaced or accumulated and Q updated.The episode terminates when a predefined truncation horizon T is reached (i.e. maximum time length of the episode). The procedure is iterated until a predefined number of events N_E is obtained. The SARSA(0) is guaranteed to convergence to an optimal action-value function for a Robbins-Monro sequence of step-sizes α_t, for further details regarding stopping criteria and convergence the reader is referred to [19]. RL approaches can tackle control problems with infinite optimisation horizon by approximating the solution with a T-stage approach. In this sense, windows of T time steps are used to truncating the time horizon, thus reducing the computational burdens [9]. The SARSA(λ) algorithm works as follows [7]:

Data: Set $e = 1$, N_E, ϵ (or a policy π to be evaluated), α, γ, λ;
Initialize $Q(s, a)$, for all $s \in S$ and $a \in A$, arbitrarily (e.g. $Q = 0$);
Initialize traces $Z(s, a) = 0$, for all $s \in S$ and $a \in A$;
while $e < N_E$ *(Episodic Loop)* **do**
    Set $t = 1$;
    Initialize starting state s_t e.g. randomly;
    Select action $a_t \in A(s_t)$ using policy derived from Q (e.g. ϵ-greedy) or $\pi(\cdot|s_t)$;
    while $t < T$ *(run an episode)* **do**
        Take action a_t, observe s_{t+1} and reward R_t;
        Select action $a_{t+1} \in A(s_{t+1})$ using policy derived from Q (e.g. ϵ-greedy) or $\pi(\cdot|s_{t+1})$;
        Compute temporal difference δ_t and update traces: $\delta_t = R_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)$;
        $Z(s_t, a_t) = Z(s_t, a_t) + 1$ (accumulate traces) or;
        $Z(s_t, a_t) = 1$ (replace traces);
        Update Q and Z for each s and a: $Q(s, a) = Q(s, a) + \alpha\delta_t Z(s, a)$;
        $Z(s, a) = \gamma\lambda Z(s, a)$;
        Set $t = t + 1$;
    end
    go to next episode $e = e + 1$;
end

Algorithm 1: The SARSA(λ) algorithm adopting replacing or accumulating eligibility traces settings.