# Towards Developing a Novel Framework for Practical PHM: a Sequential Decision Problem solved by Reinforcement Learning and Artificial Neural Networks

Luca Bellani[1], Michele Compare[2], Piero Baraldi[3], and Enrico Zio[4]

[1,2,4] *Aramis S.r.l., Italy*
*luca.bellani@aramis3d.com*
*michele.compare@polimi.it*
*enrico.zio@polimi.it*

[1,2,3] *Energy Department, Politecnico di Milano, Italy*
*piero.baraldi@polimi.it*

[4] *MINES ParisTech, PSL Research University, CRC, Sophia Antipolis, France*

[4] *Eminent Scholar, Department of Nuclear Engineering, College of Engineering, Kyung Hee University, Republic of Korea*

**ABSTRACT**

The heart of prognostics and health management (PHM) is to predict the equipment degradation evolution and, thus, its Remaining Useful Life (RUL). These predictions drive the decisions on the equipment Operation and Maintenance (O&M), and these in turn influence the equipment degradation evolution itself. In this paper, we propose a novel PHM framework based on Sequential Decision Problem (SDP), Artificial Neural Networks (ANNs) and Reinforcement Learning (RL), which allows properly considering this feedback loop for optimal sequential O&M decision making. The framework is applied to a scaled-down case study concerning a real mechanical equipment equipped with PHM capabilities. A comparison of the proposed framework with traditional PHM is performed.

## 1. INTRODUCTION

Predictive Maintenance makes use of the predictions of the equipment Remaining Useful Life (RUL) for setting efficient, just-in-time and just-right maintenance: the right part is provided to the right place at the right time, handled by the right crew. This brings big opportunities, because it allows maximizing production profits and minimizing costs and losses ((Pipe, 2009)).

A very wide range of algorithms for RUL estimation have been developed (e.g., overviewed in (Simes, Gomes, & Yasin, 2011; Jardine, Lin, & Banjevic, 2006; Dragomir, Gouriveau, Dragomir, Minca, & Zerhouni, 2014; Zio, 2012; Liu et al., 2018)), with many successful applications reported in the literature (e.g., see (Kwon, Hodkiewicz, Fan, Shibutani, & Pecht, 2016) for an overview). In these works, however, the RUL predictions are performed without considering the dynamic management of the equipment and its effects on the equipment future degradation evolution. For example, consider the prediction of the RUL for a mechanical system influenced by the applied loading conditions (e.g., pumps of the process industry (Compare, Marelli, Baraldi, & Zio, 2018), gas turbines in the energy industry (Hanachi, Mechefske, Liu, Banerjee, & Chen, 2018), aeronautic systems (Rodrigues, Yoneyama, & Nascimento, 2012; Camci, Medjaher, Atamuradov, & Berdinyazov, 2018), etc.), which in turn depend on the Operation and Maintenance (O&M) decisions taken in time for optimal equipment usage. When predicting the RUL, these future conditions of equipment usage are generally assumed constant (e.g., (Camci et al., 2018; Van Horenbeek & Pintelon, 2013; Sankararaman, Ling, Shantz, & Mahadevan, 2011; Cadini, Zio, & Avram, 2009; Cadini, Sbarufatti, Corbetta, & Giglio, 2017; Leser et al., 2017)) or behaving according to some known exogenous stochastic process (e.g., (Ling & Mahadevan, 2012; Sankararaman, 2015; Ding, Tian, Zhao, & Xu, 2018)). This does not reflect reality and the RUL

predictions that guide the O&M decisions are deemed to be incorrect.

Prediction of the future behavior of the equipment must, then, necessarily consider its intertwined relation with decisions on its O&M . To do this, we introduce a novel PHM framework, in which prognostics is framed within the Sequential Decision Problem (SDP) paradigm and we use Reinforcement Learning (RL, (R. S. Sutton & Barto, 1998; Kaelbling, Littman, & Moore, 1996; Szepesvári, 2010)) and Artificial Neural Networks (ANN, , (Ripley, 2007; Haykin, Haykin, Haykin, & Haykin, 2009; Benardos & Vosniakos, 2007)) for its solution. RL is a machine learning technique suitable for addressing SDPs (Kaelbling et al., 1996) and widely applied to decision-making problems in diverse industrial sectors, such as electricity market power transmission networks (Kuznetsova et al., 2013; Rahimiyan & Mashhadi, 2010), military trucks management (Barde, Yacout, & Shin, 2016), traffic signal control (Khamis & Gomaa, 2014; Walraven, Spaan, & Bakker, 2016), process industry (Aissani, Beldjilali, & Trentesaux, 2009), supply chain and inventory management (Wang & Usher, 2005; Keizer, Teunter, & Veldman, 2017; Pontrandolfo, Gosavi, Okogbaa, & Das, 2002; Giannoccaro & Pontrandolfo, 2002; Kim, Jun, Baek, Smith, & Kim, 2005), to cite a few.

Finally, in (Rocchetta, Compare, Patelli, & Zio, 2018; Rocchetta, Bellani, Compare, Zio, & Patelli, 2019), we have used RL to optimize the O&M of a power grid whose elements are equipped with PHM capabilities. However, the RUL knowledge is not exploited therein; rather, the PHM just tracks the degradation state of the components, whereas the dependence of the degradation evolution on the working conditions is not considered. Yet, in (Compare, Bellani, Cobelli, & Zio, 2018), we have applied RL to optimize the part flow of gas turbines not equipped with PHM capabilities, to save inventory expenses within a scheduled maintenance paradigm.

Although there are tabular dynamic programming algorithms that theoretically allow finding the exact solution of the SDP ((R. S. Sutton & Barto, 1998; Szepesvári, 2010)), the computational burden they require is not compatible with realistic PHM applications to complex systems. For this, we resort to model-free RL based on Artificial Neural Networks (ANNs, (Ripley, 2007),(Haykin et al., 2009)) to find an approximate solution.

ANNs are information processing systems composed of simple processing elements (nodes) linked by weighted connections. Inspired by the function of human brain, they have been applied to a huge variety of engineering problems (e.g., (Ripley, 2007; Haykin et al., 2009; Benardos & Vosniakos, 2007)).

For PHM, RL considers the O&M decision taken at the present time optimal only if an optimal decision will be taken also at the next decision time, considering that the equipment degradation state at that time will be influenced also by the current O&M decision. By iteratively applying this reasoning along the time horizon, one gets the sequence of decisions generating the expected maximum profit and the resulting expected degradation path.

The SDP formulation and the solution framework proposed in this paper are applied to a scaled-down case study that shows that the O&M policy found by RL outperforms any other experience-based policy.

The structure of the paper is as follows. In Section 2, we introduce the mathematical formulation of the problem. In Section 3, details about the RL algorithm are provided. In Section 4, the case study concerning a mechanical structure is considered. Results are discussed in Section 5. Finally, conclusions are drawn in Section 6.

## 2. PROBLEM SETTING

Consider a system that can be operated at $L$ different loading levels $l \in \Lambda = \{1, ..., L\}$. These are associated to the operating performance values $F_1, \ldots, F_L$ (e.g., the production rate, the absorbed load, etc.), such that $F_l \leq F_{l+1}$, $l = 1, \ldots, L-1$.

The evolution of the system degradation is described by the stochastic process $x(t) \in [\xi, \chi]$ ((R. S. Sutton & Barto, 1998; Sigaud & Buffet, 2013)), where $\chi$ is the threshold upon which the system fails, whereas $\xi$ is the minimum detectable level of $x(t)$.

The stochastic degradation process obeys the Markov property: the knowledge of the degradation state at time $t$ is sufficient to predict its evolution from $t$ on, independently on the past history that has led the system into $x(t)$. This condition is not limiting in practice, as it holds for many stochastic processes, including Brownian motion with drift, compound Poisson process, gamma process, etc. ((van Noortwijk, 2009)), which are widely used to describe degradation in a variety of domains (e.g., dikes (Speijker, Van Noortwijk, Kok, & Cooke, 2000), nuclear industry ((Baraldi et al., 2012)), power systems ((Lisnianski, Elmakias, Laredo, & Ben Haim, 2012)), mining industry (Banjevic & Jardine, 2006), to cite a few). In particular, the Markov property suits most of the crack propagation models used for PHM of mechanical systems (Sankararaman et al., 2011). Moreover, RL algorithms for semi-Markov processes are available, which require additional computational effort (e.g., (R. Sutton et al., 1999)).

We assume that the speed of the degradation process depends on the production level $l$, set by the O&M decision maker: the larger its value, the larger the revenues, the faster the degradation mechanism. This entails that we must define for every level $l \in \Lambda$ a failure time $T_f^l$, which is the random variable $T_f^l = \{\inf t \geq 0 : x(t) \geq \chi; l \in \Lambda\}$, where $E(T_f^l) \geq E(T_f^{l+1})$, $l = 1, \ldots, L-1$. Accordingly, every $\Delta t$ units of time, PHM provides $L$ different RUL predictions $\mathbf{P}(t = k\Delta t) = \mathbf{P}_k = [P_{k|1}, \ldots, P_{k|L}]$, where $P_{k|l}$ is the RUL estimated assuming that the component will continue to work at level $l$ until failure, $k \in N_0$. To be realistic, we also

assume that RULs are predicted based on signal values $y_k$, which taints $x_k$ with a random noise, $k \in N_0$.
The external demand of equipment service evolves according to a known stochastic process $d(t)$. We assume that this is described by a cyclo-stationary distribution of period $\tau$ ((Enserink & Cochran, 1994; Gardner & Spooner, 1994)), such as $E(d(t)) = E(d(t + k\tau))$, $k \in N$. For example, we can set $\tau = 1$ day to model a demand with daily periodicity. To ease the notation, we indicate by $d_k$ the demand at time $t = k\Delta t$ (i.e., $d_k = d(t = k \cdot \Delta t)$ ). Similarly, $x_k$ is the degradation at time $t = k\Delta t$.
Within the SDP framework, given the current system state, a O&M decision (i.e., action) is taken, which yields a reward and leads the system to experience a stochastic transition toward a new state. Thus, to frame PHM as a SDP (R. S. Sutton & Barto, 1998), we need to give a formal definition of the state space, actions, transitions and rewards.

### 2.1. State space

At time $t = k \cdot \Delta t$, we define the state vector $\mathbf{S}_k \in R^{L+2}$, whose $j$-th element is:

$$\mathbf{S}_{k,j} = \begin{cases} P_{k|j} & \text{if} \quad j \in \{1, ..., L\} \\ k \cdot \Delta t \mod \tau & \text{if} \quad j = L + 1 \\ d_k & \text{if} \quad j = L + 2 \end{cases} \tag{1}$$

In words, the $l$-th entry $l \in \{1, \ldots, L\}$ of the state vector defines the RUL estimated for a system working at constant level $l$. The $L + 1$-th entry re-scales the current decision time with respect to the period of the cyclo-stationary distribution, $\tau$. The last entry, $d_k$, points to the external demand of equipment service.
Notice that the values $P_{k|j}$, $j \in \{1, \ldots, L\}$ do not represent the actual RUL of the system, as this will continuously change its operating level depending on the O&M decisions.
Notice also that additional variables related to the environment can be included in the state space, which need to obey a known Markovian stochastic process. For simplicity of illustration of the modeling framework, these are disregarded in this work.

### 2.2. Actions

The decisions concern both the component operation and its maintenance. The former define the level of equipment service $F_l$, $l \in \Lambda$, whereas the latter concern two alternatives: Preventive Maintenance (PM) actions, which are performed before component failure, and Corrective Maintenance (CM) actions, performed upon failure. The corresponding downtimes, $\Pi$ and $\Gamma$, respectively, are random variables obeying Probability Density Functions (PDFs) $f_\Pi(t)$ and $f_\Gamma(t)$, respectively. To fulfill the Markov property, we assume that these distributions are exponential. The downtime of a PM action is expected to be shorter than that for CM, as the value of PHM lies in that it enables performing timely arranged preventive actions, for which all the maintenance logistic support issues have already been addressed.
In an opportunistic view, we assume that both preventive and corrective maintenance actions restore the component to an As Good As New (AGAN) state with respect to its degradation process (i.e., $x(t) = \xi$ if the maintenance action ends at time $t$).
The available O&M decisions are organized in vector $\mathbf{a} = [a_1, \ldots, a_{L+2}]$, where $l = 1, \ldots, L$ refers to setting the system at operating level $l$, whereas the last two actions correspond to the decisions of preventively maintaining the system and performing corrective actions upon failure.
The action taken at time $t = k \cdot \Delta t$ is formally indicated by vector $\mathbf{z}_k$, which encodes the binary variables $z_{k,l}$, $l \in \{1, \ldots, L + 2\}$.

$$z_{k,l} = \begin{cases} 1 & \text{if action } a_l \text{ is taken at time } t = k \cdot \Delta t \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$\sum_{l=1}^{L+2} z_{k,l} = 1 \tag{3}$$

$$z_{k-1,L+1} = 1 \quad \& \quad s_\Pi > \Delta t \Rightarrow z_{k,L+1} = 1; \tag{4}$$

$$z_{k-1,L+1} = 1 \quad \& \quad s_\Pi < \Delta t \Rightarrow y_k = \xi \tag{5}$$

$$z_{k,L+2} = 1 \Leftrightarrow y_k \geq \chi \quad \& \quad s_\Gamma > \Delta t; \tag{6}$$

$$z_{k-1,L+2} = 1 \quad \& \quad s_\Gamma < \Delta t \Rightarrow y_k = \xi \tag{7}$$

Equations 4-5 indicate that the preventive action ends when a sample $s_\Pi > \Delta t$ is drawn from $f_\Pi$, whereas Equations 6-7 indicate that the corrective maintenance action can be taken at time $t = k \cdot \Delta t$ only if the system is failed and this remains under maintenance till a sample $s_\Gamma > \Delta t$ is drawn from $f_\Gamma$. Upon maintenance, for simplicity we set the degradation to its minimum detectable level $\xi$.
For simplicity, the action taken at time $t = k \cdot \Delta t$ is also indicated by $A_k = \langle \mathbf{a}, \mathbf{z}_k \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the standard inner product in $R^{L+2}$.

### 2.3. State Stochastic Transitions

A policy $\pi$ is a mapping from the state space to the action space: every state is associated to an action, which is taken at every decision time $t = k \cdot \Delta t$. Upon the action implementation, a state transition occurs, which leads the system into state $\mathbf{S}_{k+1}$. These transitions are stochastic: both the RUL and the demand level at the next time $t = (k + 1) \cdot \Delta t$ are affected by aleatory uncertainty.
With respect to the demand level, this will evolve according to the stochastic process $d(t)$, which depends on the current time, only.
To embed the prognostic algorithm within the RL paradigm,

we need to estimate the RUL value at time $t = (k+1) \cdot \Delta t$ given that action $A_k$ has been taken at time $t = k \cdot \Delta t$. For simplicity, we assume that a model-based prognostic algorithm is available (e.g., particle filtering (Arulampalam, Maskell, Gordon, & Clapp, 2002)), whereby we can track $x_k$ (e.g., crack length) and estimate its value at the next time instant for all loading levels $l \in \Lambda$ (e.g., (Sankararaman et al., 2011; Compare, Marelli, et al., 2018; Cadini et al., 2009)). Then, we can either rely on the same degradation model to predict the RUL for a system in $x_{k+1}$ or on a simpler stochastic model that maps $x$ into the probability of failure over time (see Appendix). The assumption of the availability of a model-based prognostic algorithm is discussed in Section 5.

### 2.4. Rewards

When $z_{k,l} = 1, l \in \Lambda$, the component is operated at level $l$ and the Decision Maker receives a reward $R_k$ given by:

$$R_k = \sum_{l=1}^{L}(z_{k,l} \cdot V_{prod} \cdot \min\{d_k, F_l\}) - \sum_{l=1}^{g_k}(z_{k,l} \cdot C_{pen} \cdot (d_k - F_l)^2) - \sum_{l=g_k+1}^{L}(z_{k,l} \cdot C_{plus} \cdot (d_k - F_l)^2) \quad (8)$$

where $g_k = \max_{l:F_l \leq d_k} l$, $V_{prod}$ is the revenue per unit of production, $C_{pen}$ is a penalty incurred if the demand requirement is not satisfied and $C_{plus}$ is a penalty due to lost production. Notice that if the component fails during operation, we assume $F_l = 0$. The differences between $d_k$ and $F_l$ are squared to make the rewards less sensitive to this gap (being these values smaller than 1, they are larger than the squares). The maintenance actions result in the costs described by Equation 9, to be summed to those in Equation 8:

$$R_k = z_{k,L+1} \cdot C_{prev} + z_{k,L+2} \cdot C_{cor} - (C_{pen} \cdot (d_k)^2) \quad (9)$$

where $C_{prev}$ and $C_{cor}$ are the costs of predictive and corrective maintenance per $\Delta t$ unit time, respectively.

### 2.5. Optimal O&M decisions

The optimal policy, $\pi^*$, is the sequence of actions $\mathbf{z}_k, k \geq 1$ that maximizes the discounted sum of future rewards

$$V = E_{\pi^*}[\sum_{k=1}^{\infty} \gamma^k R_k] \quad (10)$$

where $\gamma \in (0,1)$ is a discount factor, which determines the net present value of the future rewards ((R. S. Sutton & Barto, 1998), (Gollier, 2002)), calculated as in Equations 8 and 9. In this respect, notice that we can set a threshold value $\Phi$ on the number of future steps, after which the contribution of the future rewards, $\gamma^\Phi \simeq 0$, can be considered negligible (e.g., $\Phi = 1000$, $\gamma = 0.98$; then $\gamma^\Phi = 1.7 \cdot 10^{-9}$).
Notice that in continuing tasks, setting $\gamma \simeq 1$ entails that the policy maximizing $V$ (Equation 10) is very close to that maximizing the average expected reward $\mathcal{R}$ (Tsitsiklis & Van Roy, 2002):

$$\mathcal{R} = \lim_{h \to \infty} E_\pi[R_h] \quad (11)$$

which is indeed the goal of the O&M decisions.
To solve this optimization problem, we rely on the RL algorithm detailed in the next Section.

## 3. ALGORITHM

In the RL framework (Figure 1), each state-action pair is assigned a value $Q_\pi(\mathbf{S}_k, A_k)$, which measures the expected return starting from state $\mathbf{S}_k$, taking action $A_k$ and thereafter following policy $\pi$ (R. S. Sutton & Barto, 1998):

$$Q_\pi(\mathbf{S}_k, A_k) = E_\pi\left[\sum_{K \geq k}(\gamma^{K-k} \cdot R_K) | \mathbf{S}_k, A_k\right] \quad (12)$$

Our procedure is detailed as follows. We estimate the value of $Q_\pi(\mathbf{S}_k, A_k)$ using a different ANN for each action. Thus, there are $L+2$ ANNs, $\mathcal{N}_1, \ldots, \mathcal{N}_{L+2}$, with network weights $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{L+2}$, respectively. Network $\mathcal{N}_l$, $l = 1, ...L+2$, receives in input the state vector $\mathbf{S}_k$ (Equation 1) and returns the approximated value $\hat{q}_l(\mathbf{S}_k|\boldsymbol{\mu}_l)$ of $Q_\pi(\mathbf{S}_k, A_k = a_l)$.
To speed up the training of the ANNs ((Riedmiller, 2005)), we initially apply a standard supervised training over a batch of relatively large size $n_{ei}$, to set weights $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{L+2}$. To collect this batch, we randomly sample the first state $\mathbf{S}_1$ and, then, move $n_{ei} + \Phi$ steps forwards by uniformly sampling from the set of applicable actions and collecting the transitions $\mathbf{S}_k, A_k \to \mathbf{S}_{k+1}, A_{k+1}$ with the corresponding rewards $R_k, k = 1, ..., n_{ei} + \Phi - 1$.
Every network $\mathcal{N}_l, l \in \{1, \ldots, L+2\}$, is trained on the set of states $\{\mathbf{S}_k | k = 1, ..., n_{ei}, A_k = l\}$ in which the $l$-th action is taken, whereas the reward that the ANN learns is the Monte-Carlo estimate $Y_k$ of $Q_\pi(\mathbf{S}_k, A_k)$:

$$Y_k = \sum_{k'=k}^{k+\Phi} \gamma^{k'-k} \cdot R_{k'} \quad (13)$$

After this initial training, we apply Q-learning (e.g., (R. S. Sutton & Barto, 1998),(Szepesvári, 2010)) to find the ANN approximation of $Q_\pi(\mathbf{S}_k, A_k)$. Namely, every time the state $\mathbf{S}_k$ is visited, the action $A_k$ is selected among all available actions according to the $\epsilon-$greedy policy $\pi$: the action with the largest value is selected with probability $1 - \epsilon$, whereas a different applicable action is sampled with probability $\epsilon$. The exploration rate, $\epsilon$, is continuously decreased to lead the algorithm to convergence. Then, the immediate reward

and the next state are observed, and weights $\boldsymbol{\mu}_{A_k}$ of network $\mathcal{N}_{A_k}$ are updated: a single run of the back-propagation algorithm is done ((Ripley, 2007),(Haykin et al., 2009)) using $R_k + \gamma \cdot \max_{l\in\{1,\dots,L+2\}} \hat{q}_l(\mathbf{S}_{k+1}|\boldsymbol{\mu}_l)$ as target value (Equation 14). This yields the following updating:

$$\boldsymbol{\mu}_{A_k} \leftarrow \boldsymbol{\mu}_{A_k} + \alpha \cdot [R_k + \gamma \cdot \max_{l\in\{1,\dots,L+2\}} \hat{q}_l(\mathbf{S}_{k+1}|\boldsymbol{\mu}_l) - \hat{q}_{A_k}(\mathbf{S}_k|\boldsymbol{\mu}_{A_k})] \cdot \nabla \hat{q}_{A_k}(\mathbf{S}_k|\boldsymbol{\mu}_{A_k}) \quad (14)$$

where $\alpha > 0$ is the value of the learning rate ((R. S. Sutton & Barto, 1998)).

Notice that the accuracy of the estimates provided by the proposed algorithm strongly depends on the frequency at which the actions are taken in every state: the larger the frequency, the larger the information from which the network can learn the state-action value (R. S. Sutton & Barto, 1998). In real industrial applications, where systems spend most of the time in states for which the estimated RUL $\mathbf{P}_k$ is relatively large ((Frank, Mannor, & Precup, 2008)), this may entail a bias or large variance in the ANN estimations of $Q_\pi(\mathbf{S}_k, A_k)$ for rarely visited states. To overcome this issue, we increase the exploration by dividing the simulation of the system, and its interactions with the environment and O&M decisions, into episodes of fixed length $W$. Thus, we run $N_{ei}$ episodes, each one entailing $W$ decisions; at the beginning of each episode, we sample the first state uniformly over all states. This procedure increases the frequency of visits to highly degraded states and reduces the estimation error. At each episode $ei \in \{1, \dots, N_{ei}\}$, we decrease the exploration rate $\epsilon = \epsilon_{ei}$ according to $\epsilon = \epsilon_0 \cdot \tau_\epsilon^{ei}$, and the learning rate $\alpha = \alpha_{ei}$ according to $\alpha_{ei} = \alpha_0 \cdot \frac{N_\alpha+1}{N_\alpha+ei}$ ((R. S. Sutton & Barto, 1998)). The algorithm is reported in Appendix 2.

Figure 1 summarizes the presented framework. The PHM system continuously monitors the component and provides the RUL $P_{k|l}$ for each operating level $l$. The RL selects the action $a_k$ based on the PHM predictions, external demand $d_k$ and re-scaled time with respect to $\tau$, i.e., with probability $1-\epsilon$, $a_k = \arg\max_{l=\{1,\dots,L+2\}} \hat{q}_l(\mathbf{S}_k|\boldsymbol{\mu}_l)$, otherwise $a_k$ is selected randomly among all actions. The component is operated according to action $a_k$ (i.e., it undergoes maintenance or works at the provided performance level). At the next decision time, network $\mathcal{N}_{a_k}$ is trained based on the immediate reward and estimate of the maximum value of the next state provided by the neural networks.

## 4. CASE STUDY

We consider a pumping system working at $L = 8$ operation levels $F_l = l$, in arbitrary units. We assume that it is affected by a fatigue crack growth degradation, modeled as in (Sankararaman et al., 2011; Zio & Compare, 2013).

The crack length $x_k$ (in mm) at time $t = k \cdot \Delta t$, $k \in N$, $\Delta t = 12$ hours, is described by the Paris-Erdogan (PE) law ((Kozin & Bogdanoff, 1989)):

$$x_k = x_{k-1} + e^{\psi_k} \cdot C_l \cdot (\beta \cdot \sqrt{x_{k-1}})^n \Delta t \quad (15)$$

where $\beta = 1$ and $n = 1.3$ are constant parameters that depend on the material properties, whereas $\psi_k, k \geq 1$ are independent and identically distributed normal random variables, i.e. $\psi_k \sim \mathcal{N}(0, \sigma_\psi^2)$, $\sigma_\psi = 1.7$ ((Kozin & Bogdanoff, 1989)). The initial crack length is $\xi = 1$ mm. The pump fails when $x_k \geq \chi = 100$ mm ((Cadini et al., 2009)).

$C_l$ is the speed of the crack propagation evolution, which depends on the performance level of the pump. For illustration, we consider that the crack propagation speed is quadratic in $l$ (Table 1) and that the values of $\beta$ and $n$ do not depend on the operating level $l$ ((Kozin & Bogdanoff, 1989)).

Table 1. Parameter $C_l$ vs different operating levels $l$

| $l=1$ | $l=2$ | $l=3$ | $l=4$ |
|---|---|---|---|
| 0.0006 | 0.001 | 0.0015 | 0.0021 |
| $l=5$ | $l=6$ | $l=7$ | $l=8$ |
| 0.0033 | 0.005 | 0.0071 | 0.0096 |

The external demand stochastic process $d_k$ is described by:

$$d_k \sim \begin{cases} \mathcal{U}(4,8) & k\Delta t \bmod \tau \in [7,19) \text{ (day demand)} \\ \mathcal{U}(0,4) & \text{otherwise (night demand)} \end{cases} \quad (16)$$

with $\tau = 1$ day (according to a simplified model of the demand derived from (Lojowska, Kurowicka, Papaefthymiou, & van der Sluis, 2012; Power & Verbic, 2017)). To be realistic, we also assume that the measured crack length $y_k$ at time $t = k \cdot \Delta t$ is affected by a random noise: $y_k = \max\{x_k + e_k, \xi\}, e_k \sim \mathcal{N}(0, 0.5)$.

Whichever the prognostic algorithm is, to avoid its running every time a decision needs to be taken, we have described the uncertainty in the RUL prediction through Weibull distributions, i.e. $P_{k|l} \sim \mathcal{W}(\alpha_l(y_k), \beta_l(y_k))$. Parameters $\alpha_l(y_k)$ and $\beta_l(y_k)$ are estimated through the procedure reported in Appendix. Notice that this simplification is aimed at reducing the computational times, only. Alternatively, the RL algorithm can learn from the RUL given by the PHM algorithm over simulated degradation paths. Notice also that the RL algorithm allows the introduction of further uncertainty in the simulation model, measurement system or prognostics algorithm, as it is model-free algorithm that directly learns from simulation.

With respect to the reward values, the parameters of Equations 8 and 9 are reported in Table 2, in arbitrary units, whereas the random durations $f_\Gamma$ and $f_\Pi$ are described by exponential distributions with mean 60 hours and 12 hours, respectively. These values are for illustration, only. The RL algorithm settings are reported in Table 3. The RL algorithm, coded in Python, converges in approximately 1h on a 3.6GHz CPU, 16MB RAM computer.

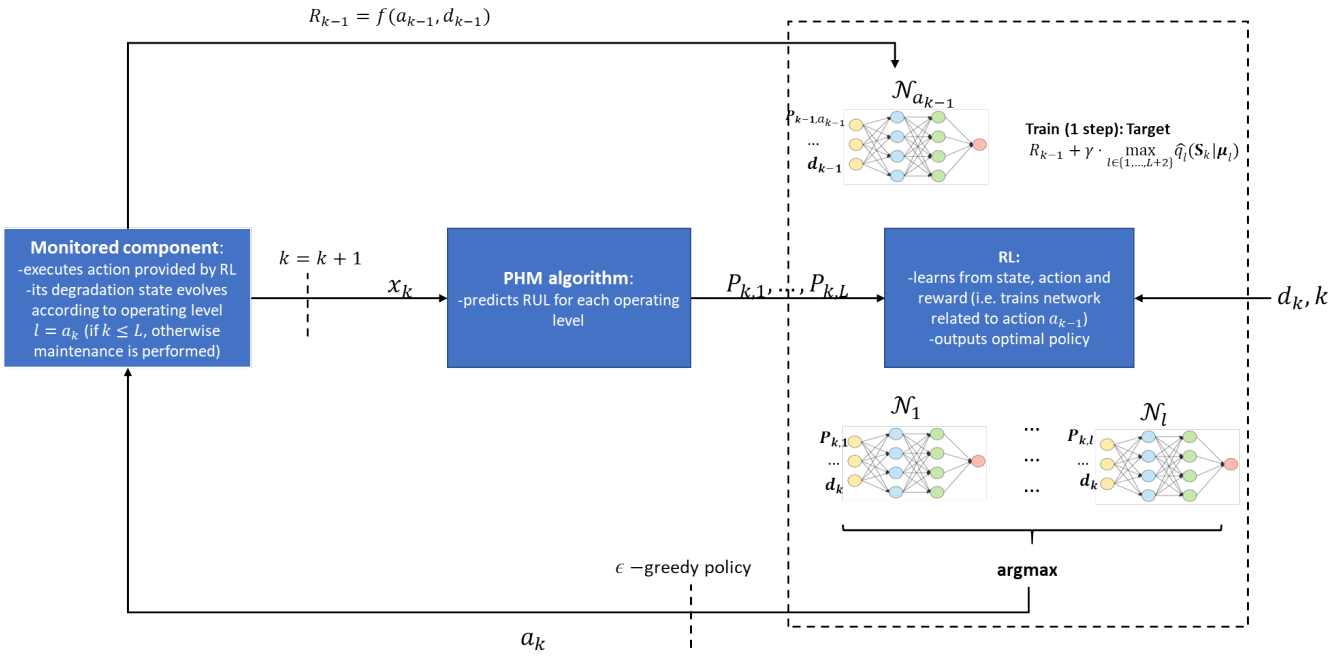


Figure 1. RL framework.

Table 2. Reward parameters

| $V_{prod}$ | $C_{pen}$ | $C_{plus}$ | $C_{cor}$ | $C_{prev}$ |
|---|---|---|---|---|
| 20 | 10 | 2 | 5 000 | 2 000 |

Table 3. RL and neural network parameters

| $N_{ei}$ | $n_{ei}$ | $\epsilon_0$ | $\tau_\epsilon$ | $\alpha_0$ | $N_\alpha$ | $W$ | $\gamma$ |
|---|---|---|---|---|---|---|---|
| 8 000 | 100 000 | 0.5 | 900 | 0.01 | 600 | 100 | 0.99 |

To detail the policy provided by RL, Figures 2-3 show the actions taken in correspondence of different demand levels overnight and during the day, respectively, versus the crack length. From the analysis of all Figures, we can see that:

- When the demand level is below 1, i.e., $d_k < 1$ (Figure 2 top-left), the component is set at operating level $l = 1$, i.e., $A_k = a_1$. Preventive maintenance ($A_k = a_9$) is performed when $y_k \geq 77$.
- When the demand level is between 1 and 2, i.e., $1 \leq d_k < 2$ (Figure 2 top-right), preventive maintenance is performed when $y_k \geq 77$, whereas the component is set at operating level $l = 1$ (i.e., $A_k = a_1$) when the measured crack length is small or close to the maintenance threshold; $A_k = a_2$ when $y_k$ is at intermediate values (approximately between 15 and 70). RL tends to take action $a_1$ in correspondence of small crack lengths because the revenues are small, whereas the loss of RUL is very large (see the highest curve in Figure 6 in Appendix). The selection of $a_1$ in correspondence of large crack values comes from the reduction of the risk of failure it yields, with larger probabilities of surviving up to the next day, in which it is more convenient to run the system at larger operating levels.
- When the demand level is between 2 and 3, i.e., $2 \leq d_k < 3$ (Figure 2 bottom-left), the component is set at operating level $l = 3$ ($A_k = a_3$) if the measured crack length is small, otherwise $A_k = a_2$ when $y_k$ is at intermediate values: the larger the demand level, the larger the convenience of setting $F_l = 3$. Coherently, the limit value between actions $a_2$ and $a_3$ is $y_k = 17$ when the demand level $d_k$ is close to 2 and $y_k = 39$ when $d_k \simeq 3$. Maintenance is performed when $y_k \geq 76$.
- When $3 \leq d_k < 4$ (Figure 2 bottom-right), the component is set at operating level $l = 3$, i.e. $A_k = a_3$, when the measured crack length is small, whereas $A_k = a_4$ when $y_k$ is at intermediate values. This is a counter-intuitive result found by RL. Maintenance is performed when $y_k \geq 77$ for values of demand level close to 3 and when $y_k \geq 84$ when $d_k \simeq 4$.
- When $4 \leq d_k < 5$ (Figure 3 top-left), the component is set at operating level $l = 4$, i.e. $A_k = a_4$, unless preventive maintenance is scheduled. Maintenance is performed when $y_k \geq 84$ for $d_k = 4$ and when $y_k \geq 97$ when $d_k \simeq 5$.

- When $5 \leq d_k < 6$ (Figure 3 top-right), the component is set at operating level $l = 5$, i.e. $A_k = a_5$, when the measured crack length is small, whereas $A_k = a_6$ at larger crack length values. Maintenance is hardly ever performed apart from values of demand close to 5. Notice that it is more profitable to perform maintenance at demand values slightly above 5 than at values slightly below 5 (Figure 3 top-left). This is due to the fact that $C_{pen}$ is larger than $C_{plus}$ and that $F_5$ does not fulfill demand values above 5.
- When $6 \leq d_k \leq 8$ (Figure 3 bottom-left and bottom-right), the component is set at operating level $l = 7$, i.e., $A_k = a_7$. Notice that when the demand value is close to its maximum 8 and the crack length is very large, $A_k = a_8$. However, this may be an error due to the very low frequency of visits to states with such a large crack depth.

To fairly evaluate the policy found by RL, we compare its expected average reward $\mathcal{R}$ with that provided by two heuristic policies, which rely on the following rules:

- Maintenance is performed when the mean of the RULs predicted for the performance levels experienced during the day (i.e., $\frac{1}{L/2}\sum_{l=L/2}^{L} P_{k|l}$) is smaller than an optimal threshold $T$. The pump is always set at the maximum level $l$ below the required demand, whereby action $A_k = a_j, j = \max\{l \in \Lambda | F_l - 1 \leq d_k\}$ is selected. This policy is referred to as "low performance".
- Maintenance is performed only overnight (i.e., at lower external demand levels), upon the achievement of an optimal threshold value $T$ on the equipment day-average RUL $\frac{1}{L/2}\sum_{l=L/2}^{L} P_{k|l}$. The pump is always set at minimum level $l$ such that the demand requirement is satisfied; formally, we select action $A_k = a_j$, $j = \min\{l \in \Lambda | F_l \geq d_k\}$. For this reason, we refer to this policy as "high performance".

In both cases, the optimal threshold $T$ has been set to obtain the maximum $\mathcal{R}$. Table 4 reports the expected reward $\mathcal{R}$ for different values of $T$, estimated through 100 000 Monte Carlo trials. The optimal threshold values for the "low performance" and "high performance" policies are $T = 132$ hours and $T = 144$ hours, respectively, which yield an average reward $\mathcal{R} = 40.93$ and $\mathcal{R} = 41.13$, respectively. These thresholds correspond to performing preventive maintenance overnight when the crack length is on average $y_k = 75$ mm for $T = 132$ hours and $y_k = 71$ mm for $T = 144$ hours. From now on, we refer to "low performance" and "high performance" policies considering the optimal threshold values. In the considered case study, the "low performance" policy is worse than the "high performance", which is over-performed by the RL policy: this yields an average reward $\mathcal{R} = 41.93$.

To provide an intuitive idea of the difference between the RL policy and the heuristic policies, we can refer to Figures 4-5, which show the actions taken by RL for every combination of demand level $d_k$ and measured crack length $y_k$ with the corresponding ones of "low performance" and "high performance" policies, respectively. As expected, the larger the demand level, the larger the load set by both RL and heuristic policies. Preventive maintenance, $a_9$, is performed at large crack length levels and mainly when the demand is low (i.e., overnight).
From Figures 4-5, we can argue that the RL policy tends to mix between the "high performance" and "low performance" policies: the operating performance $F_l$ is set just above or just below the required demand, trying to remain as close as possible to the demand. Preventive maintenance is performed at values close to those of the two optimized heuristic policies. However, in some cases RL performs maintenance during the day. Moreover, the larger the demand value, the larger the crack length at which RL sets maintenance, whereas this is fixed for the two heuristics.

Finally, notice that the tabular approach is not doable to address the presented case study, whereby it is not possible to compare the found solution to a ground-truth reference value, unless we over-simplify the case study. To see this, we can analyze what happens when we consider a rough discretization of the state space. Namely, Figure 7 shows the estimated RUL distributions at different crack lengths for the considered operating levels; from this, we can see that the maximum RUL level (i.e., at $l = 1$ and crack length $x_k = 1$) is about $\overline{R} = 6000$ hours. Now, if the possible RUL values in $(0, \overline{R}]$ are partitioned into equally spaced intervals, say, of length 12 hours, there are $\frac{6000}{12} = 500$ different RUL levels that can be provided by each PHM system. Of course, there are 2 possible different decision periods (i.e, day and night). If we further assume that the demand level can only take values equally spaced of 0.5 in $[0.5, 8]$ (i.e, $D = 16$ possible values), then the state space dimension turns out to reach $500^8 \cdot 16 \cdot 2 \simeq 10^{23}$. This space is not computationally tractable.

## 5. DISCUSSION

From the analysis presented above, some important issues arise when applying RL to the PHM context, which deserve a discussion. To do this, we compare the RUL-based PHM with its "static" counterpart (Table 5).
As mentioned above, a variety of prognostic algorithms have been developed, which allow exploiting the different knowledge, information and data ((Zio, 2016)) that can be available to predict the RUL. On the contrary, RL requires a fine-tuned model-based algorithm (Section 2.3). This entails that when only data-driven algorithms are available for prognostics (e.g., (Cannarile et al., 2018)), we cannot straightforwardly apply the RL framework, as we cannot simulate the RULs that we will predict at the next decision instants. The

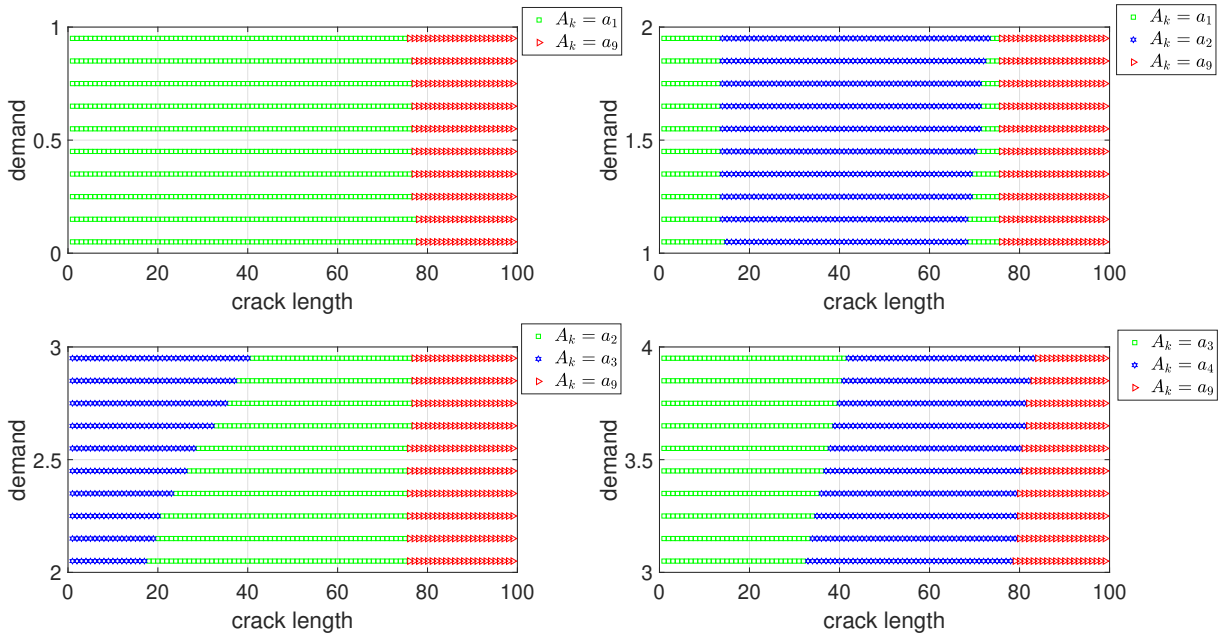


Figure 2. Action taken vs demand and measured crack length for RL policy overnight. Subfigures 1-4 refer to demand level in $[0, 1]$, $[1, 2]$, $[2, 3]$, $[3, 4]$, respectively.

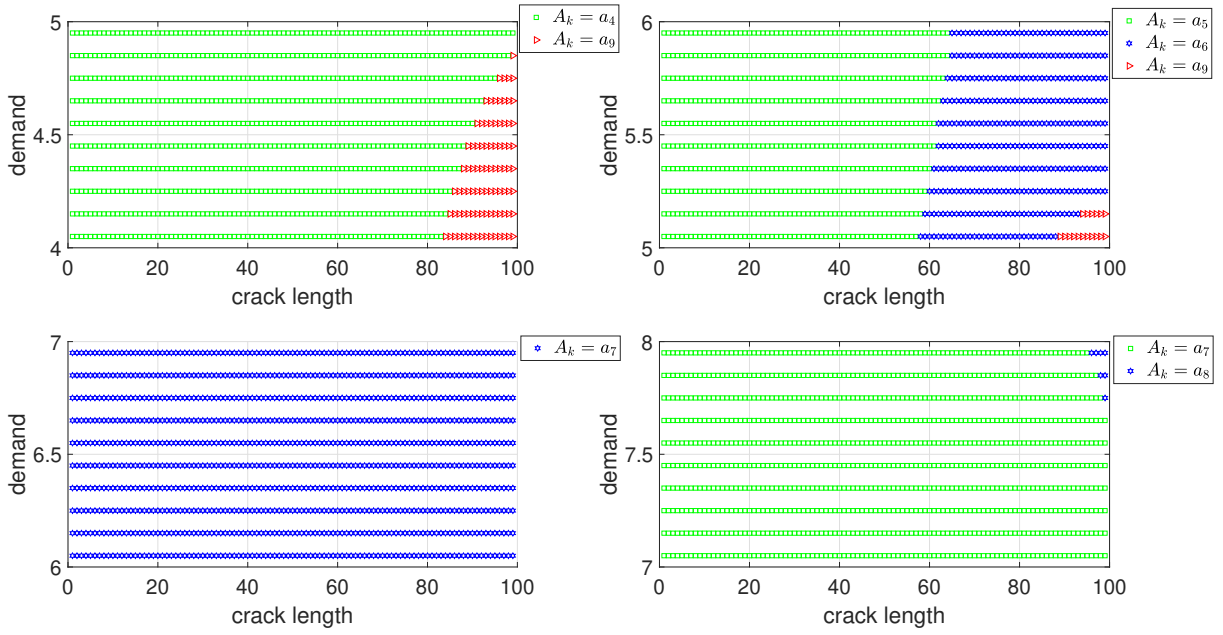


Figure 3. Action taken vs demand and measured crack length for RL policy during the day. Subfigures 1-4 refer to demand level in $[4, 5]$, $[5, 6]$, $[6, 7]$, $[7, 8]$, respectively.

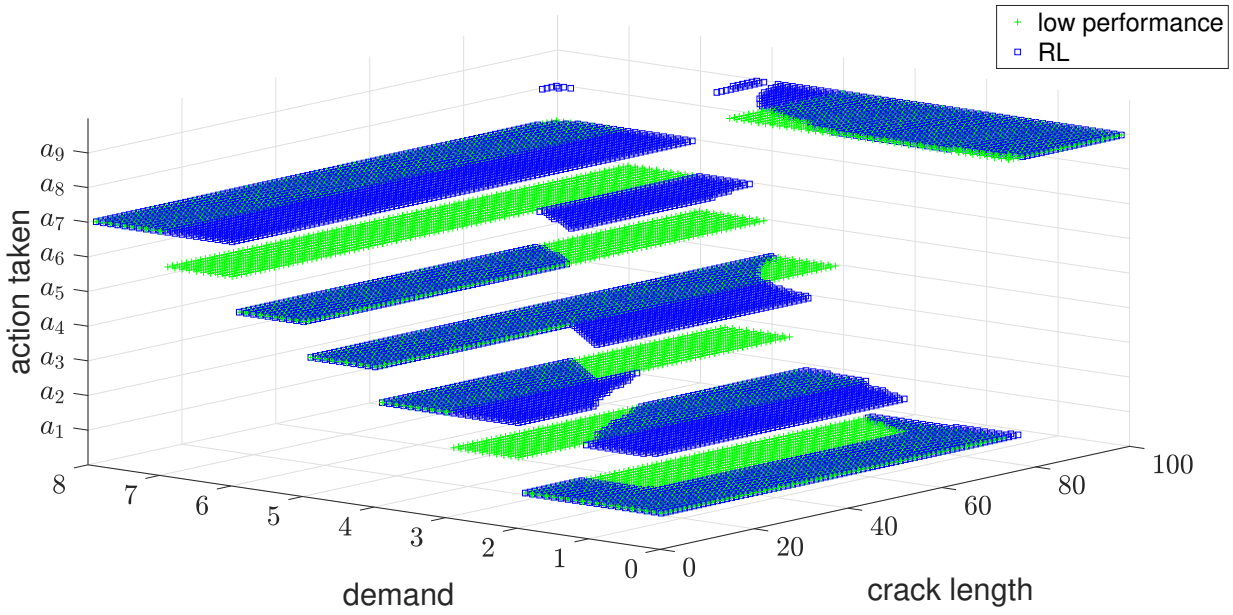


Figure 4. Action taken vs demand and measured crack length for "low performance" and RL policy. The optimal threshold $T = 132$ hours corresponds to $y_k = 71$ if maintenance is performed overnight, i.e., at demand level $d_k \leq 4$.

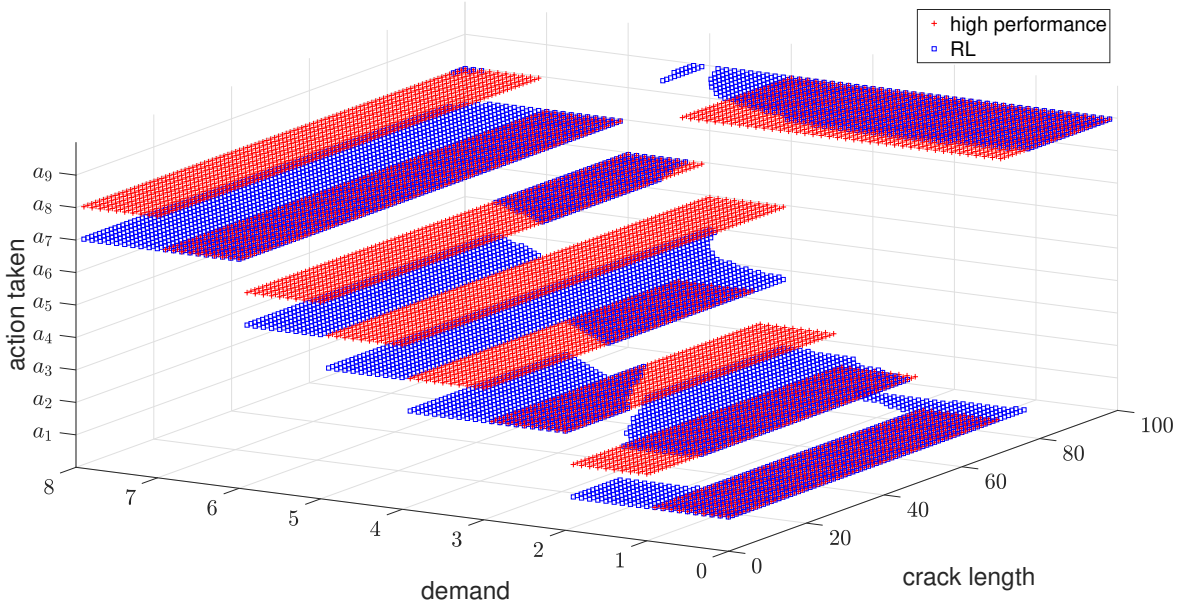


Figure 5. Action taken vs demand and measured crack length for "high performance" and RL policy. The optimal threshold $T = 144$ hours corresponds to $y_k = 75$ if maintenance is performed overnight, i.e., at demand level $d_k \leq 4$.

Table 4. $\mathcal{R}$ vs different RUL threshold values $T$ (in hours) for the two different heuristic policies

| Threshold (T) | 60 | 72 | 84 | 96 | 108 | 120 | 132 | 144 | 156 | 168 | 180 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Low performance | 35.53 | 38.1 | 39.56 | 40.5 | 40.81 | 40.82 | 40.93 | 40.90 | 40.49 | 40.04 | 39.66 |
| High performance | 32.06 | 35.29 | 37.25 | 38.62 | 40.12 | 40.81 | 41.09 | 41.13 | 41.00 | 40.71 | 40.36 |

extension of RL-PHM to these cases is a fundamental open research question, which can be addressed by building on the metrics for prognostics introduced in (Saxena, Celaya, Saha, Saha, & Goebel, 2010a, 2010b). With these, we can estimate the distribution of the RUL values at any time instant, based on the expected time-variant performances (e.g., accuracy, false positive rate, etc.) of the PHM algorithm, whichever the algorithm is (see (Compare, Bellani, & Zio, 2017; Compare, Bellani, & Zio, 2017)). This issue will be addressed in future research works.

With reference to the second line of Table 5, we can notice that differently from the mainstream applications of RL to robots ((Kober, Bagnell, & Peters, 2013)), gaming ((Silver et al., 2016)), etc., when it is applied to PHM, RL cannot learn from direct interaction with the environment, as this would require unprofitably operating a large number of systems. On the other hand, as highlighted in (Compare, Baraldi, & Zio, 2018), also the static PHM suffers from the practical difficulty of learning from direct interaction with the environment: gathering time-to-failure trajectories is not doable for critical systems, which are conservatively operated to avoid failures. Then, a realistic simulator of the state evolution depending on the actions taken is required in both cases. This seems not a limiting point in the Industry 4.0 era, when digital twins are more and more common and refined. Notice that this issue is closely related to that of the applicability of RL to data-driven settings, discussed in the previous paragraph: a realistic simulator is expected to encode sound models of the degradation mechanisms affecting the monitored system.

We have proved through a simple case study that the RL policy yields on average better results than the two considered good-sense O&M policies; larger benefits are expected from RL application to more complex case studies. This is due to the fact that RL provides in output the optimal O&M policy, instead of the RUL for the different loading conditions (Table 5, third and fourth rows). Whilst it goes without saying that the knowledge of the RUL is per se valuable, as it enables setting efficient maintenance actions, however from the proposed framework it clearly arises that this is limiting with respect to the full potential of PHM (Table 5, last row). In this respect, notice that the RUL values predicted by any PHM algorithm are "correct" only for the single decision period. After that, a new prediction is available, which holds for the next interval only. Then, within the RL framework the health management resulting from prognostics amounts to an optimal sequence of actions driven by the RULs that are expected to be estimated in the future, also in consequence of the future actions. The actual RUL is never known, and neither it can be estimated a-posteriori: the optimal actions avoid system failure, whereby the time to failure goes to infinite. This questions the concept of actual RUL itself.

Finally, the RL solutions are difficult to understand. For example, as mentioned before the results of Figure 2 bottom-right are very counter-intuitive: we would expect they be similar to those of Figure 2 bottom-left. Then, the application of RL for O&M decision requires asset managers to take actions with weak awareness of their optimality. In this respect, on one side we should bear in mind that it is quite easy to check whether the RL policy outperforms those currently used to manage the system. On the other side, we can corroborate the RL framework with interactive simulators, through which the asset manager can take actions different from those proposed by RL to understand why they lead to counter-intuitive results. Further research work will focus on the integration of the developed framework with an interactive simulator and test slight changes to the RL policy to assess the optimality and the robustness of the proposed framework.

Further research work will focus on the integration of the developed framework with an interactive simulator, for testing slight changes to the RL policy and assess the optimality and robustness of the proposed framework.

Finally, the novel RL framework has been demonstrated by application to a scaled-down case study. Computational and modeling issues will arise when applying this framework to more complex industrial issues. In this respect, future research will exploit the recent advancements in both RL algorithms development (e.g., double Q-learning, batch training with experience replay (Silver et al., 2016; Mnih et al., 2013, 2015)) and computing architectures (e.g., (Mnih et al., 2016))).

Based on the considarations above, it seems fair to conclude that the proposed RL algorithm paves the way to research paths that are fundamental for the full development of PHM for practical applications.

## 6. CONCLUSIONS

In this work, a novel framework for practical PHM has been proposed, which allows overcoming the limitation of the current practice of PHM of predicting the degradation evolution and RUL independently on the O&M actions that are taken to maximize the equipment profitability, and which inevitably influence degradation and RUL. The framework is based on SDP and its solution on RL and ANN.

For illustration, a scaled-down case study has been worked out within this new framework, to highlight the benefits for

Table 5. Comparison of the proposed PHM framework with traditional PHM

| | Traditional PHM | Proposed PHM |
|---|---|---|
| Information required to develop the algorithms | Monitoring data relevant to different loading conditions to develop data-driven prognostic algorithms.<br>Physical model of degradation to build model-based algorithm.<br>Hybrid approaches available. | Physical model of degradation and stochastic process describing the environment behavior. Both must be Markovian. Semi-Markov extensions are available at the price of larger computational times. |
| On-line application requirements | Input: Signal Data.<br>Model: Trained on time-to-failure trajectories. | Input: Signal Data and Actions.<br>Model: Trained on simulated state-action-rewards trajectories |
| Output | RUL for every loading condition | Optimal Decision Policy |
| Strength | Timely performed actions | Optimal O&M |
| Weakness | Decisions based on RULs relevant to static loading conditions: sub-optimal decisions | Optimal decision policy may be difficult to understand |

operating a PHM-equipped system.
Some open issues have been highlighted, which need to be addressed for a wide and effective application of PHM to industrial practice. These will be the focus of future research work.

## REFERENCES


Aissani, N., Beldjilali, B., & Trentesaux, D. (2009). Dynamic scheduling of maintenance tasks in the petroleum industry: A reinforcement approach. *Engineering Applications of Artificial Intelligence*, *22*(7), 1089–1103.

Arulampalam, M., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, *50*(2), 174-188. doi: 10.1109/78.978374

Banjevic, D., & Jardine, A. (2006). Calculation of reliability function and remaining useful life for a markov failure time process. *IMA Journal of Management Mathematics*, *17*(2), 115-130. doi: 10.1093/imaman/dpi029

Baraldi, P., Balestrero, A., Compare, M., Zio, E., Benetrix, L., & Despujols, A. (2012). A new modeling framework of component degradation. *Advances in Safety, Reliability and Risk Management - Proceedings of the European Safety and Reliability Conference, ESREL 2011*, 776-781.

Barde, S. R., Yacout, S., & Shin, H. (2016). Optimal preventive maintenance policy based on reinforcement learning of a fleet of military trucks. *Journal of Intelligent Manufacturing*, 1–15.

Benardos, P., & Vosniakos, G.-C. (2007). Optimizing feedforward artificial neural network architecture. *Engineering Applications of Artificial Intelligence*, *20*(3), 365–382.

Cadini, F., Sbarufatti, C., Corbetta, M., & Giglio, M. (2017). A particle filter-based model selection algorithm for fatigue damage identification on aeronautical structures. *Structural Control and Health Monitoring*, *24*(11). doi: 10.1002/stc.2002

Cadini, F., Zio, E., & Avram, D. (2009). Monte carlo-based filtering for fatigue crack growth estimation. *Probabilistic Engineering Mechanics*, *24*(3), 367-373. doi: 10.1016/j.probengmech.2008.10.002

Camci, F., Medjaher, K., Atamuradov, V., & Berdinyazov, A. (2018). Integrated maintenance and mission planning using remaining useful life information. *Engineering Optimization*, 1–16.

Cannarile, F., P., B., Compare, M., Borghi, D., Capelli, L., & Zio, E. (2018). *A heterogeneous ensemble approach for the prediction of the remaining useful life of packaging industry machinery* (B. A. E. G. C. v. E. K. T. E. V. J. E. S. Haugen S. (Ed.) & C. P. Reliability Safe Societies in a Changing World, Eds.). London.

Compare, M., Baraldi, P., & Zio, E. (2018). Predictive maintenance in the industry 4.0. *Journal of Quality and Maintenance Engineering*. (Submitted)

Compare, M., Bellani, L., Cobelli, E., & Zio, E. (2018). Reinforcement learning-based flow management of gas turbine parts under stochastic failures. *The International Journal of Advanced Manufacturing Technology*, *99*(9-12), 2981–2992.

Compare, M., Bellani, L., & Zio, E. (2017, June). Avail-

ability model of a phm-equipped component. *IEEE Transactions on Reliability*, *66*(2), 487-501. doi: 10.1109/TR.2017.2669400

Compare, M., Bellani, L., & Zio, E. (2017). Reliability model of a component equipped with phm capabilities. *Reliability Engineering & System Safety*, *168*, 4 - 11. (Maintenance Modelling) doi: https://doi.org/10.1016/j.ress.2017.05.024

Compare, M., Marelli, P., Baraldi, P., & Zio, E. (2018). 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*. (Article in Press) doi: 10.1177/1748006X18757077

Ding, F., Tian, Z., Zhao, F., & Xu, H. (2018). An integrated approach for wind turbine gearbox fatigue life prediction considering instantaneously varying load conditions. *Renewable Energy*, *129*, 260-270. doi: 10.1016/j.renene.2018.05.074

Dragomir, O., Gouriveau, R., Dragomir, F., Minca, E., & Zerhouni, N. (2014). Review of prognostic problem in condition-based maintenance. *2009 European Control Conference, ECC 2009*, 1587-1592.

Enserink, S., & Cochran, D. (1994). A cyclostationary feature detector. In *Proceedings of 1994 28th asilomar conference on signals, systems and computers* (pp. 806–810).

Frank, J., Mannor, S., & Precup, D. (2008). Reinforcement learning in the presence of rare events. In *Proceedings of the 25th international conference on machine learning* (pp. 336–343).

Gardner, W. A., & Spooner, C. M. (1994). The cumulant theory of cyclostationary time-series. i. foundation. *IEEE Transactions on signal processing*, *42*(12), 3387–3408.

Giannoccaro, I., & Pontrandolfo, P. (2002). Inventory management in supply chains: a reinforcement learning approach. *International Journal of Production Economics*, *78*(2), 153–161.

Gollier, C. (2002). Time horizon and the discount rate. *Journal of economic theory*, *107*(2), 463–473.

Hanachi, H., Mechefske, C., Liu, J., Banerjee, A., & Chen, Y. (2018). Performance-based gas turbine health monitoring, diagnostics, and prognostics: A survey. *IEEE Transactions on Reliability*. (Article in Press) doi: 10.1109/TR.2018.2822702

Haykin, S. S., Haykin, S. S., Haykin, S. S., & Haykin, S. S. (2009). *Neural networks and learning machines* (Vol. 3). Pearson Upper Saddle River.

Jardine, A., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, *20*(7), 1483-1510. doi: 10.1016/j.ymssp.2005.09.012

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, *4*, 237–285.

Keizer, M. C. O., Teunter, R. H., & Veldman, J. (2017). Joint condition-based maintenance and inventory optimization for systems with multiple components. *European Journal of Operational Research*, *257*(1), 209–222.

Khamis, M. A., & Gomaa, W. (2014). Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework. *Engineering Applications of Artificial Intelligence*, *29*, 134–151.

Kim, C., Jun, J., Baek, J., Smith, R., & Kim, Y. (2005). Adaptive inventory control models for supply chain management. *The International Journal of Advanced Manufacturing Technology*, *26*(9-10), 1184–1192.

Kober, J., Bagnell, J., & Peters, J. (2013). Reinforcement learning in robotics: A survey. *International Journal of Robotics Research*, *32*(11), 1238-1274. doi: 10.1177/0278364913495721

Kozin, F., & Bogdanoff, J. (1989). Probabilistic models of fatigue crack growth: Results and speculations. *Nuclear Engineering and Design*, *115*(1), 143-171. doi: 10.1016/0029-5493(89)90267-7

Kuznetsova, E., Li, Y.-F., Ruiz, C., Zio, E., Ault, G., & Bell, K. (2013). Reinforcement learning for microgrid energy management. *Energy*, *59*, 133–146.

Kwon, D., Hodkiewicz, M., Fan, J., Shibutani, T., & Pecht, M. (2016). Iot-based prognostics and systems health management for industrial applications. *IEEE Access*, *4*, 3659-3670. doi: 10.1109/ACCESS.2016.2587754

Leser, P., Hochhalter, J., Warner, J., Newman, J., Leser, W., Wawrzynek, P., & Yuan, F.-G. (2017). Probabilistic fatigue damage prognosis using surrogate models trained via three-dimensional finite element analysis. *Structural Health Monitoring*, *16*(3), 291-308. doi: 10.1177/1475921716643298

Ling, Y., & Mahadevan, S. (2012). Integration of structural health monitoring and fatigue damage prognosis. *Mechanical Systems and Signal Processing*, *28*, 89-104. doi: 10.1016/j.ymssp.2011.10.001

Lisnianski, A., Elmakias, D., Laredo, D., & Ben Haim, H. (2012). A multi-state markov model for a short-term reliability analysis of a power generating unit. *Reliability Engineering and System Safety*, *98*(1), 1-6. doi: 10.1016/j.ress.2011.10.008

Liu, Z., Jia, Z., Vong, C.-M., Han, J., Yan, C., & Pecht, M. (2018). A patent analysis of prognostics and health management (phm) innovations for electrical systems. *IEEE Access*, *6*, 18088-18107. doi: 10.1109/ACCESS.2018.2818114

Lojowska, A., Kurowicka, D., Papaefthymiou, G., & van der Sluis, L. (2012). Stochastic modeling of power demand due to evs using copula. *IEEE Transactions on Power Systems*, *27*(4), 1960–1968.

Mnih, V., Badia, A., Mirza, L., Graves, A., Harley, T., Lillicrap, T., ... Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In (Vol. 4, p. 2850-2869).

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A., Veness, J., Bellemare, M., ... Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529-533.

Pipe, K. (2009). Practical prognostics for condition based maintenance. *Annual Forum Proceedings - AHS International*, *2*, 1670-1679.

Pontrandolfo, P., Gosavi, A., Okogbaa, O. G., & Das, T. K. (2002). Global supply chain management: a reinforcement learning approach. *International Journal of Production Research*, *40*(6), 1299–1317.

Power, T., & Verbic, G. (2017). A nonparametric bayesian model for forecasting residential solar generation. In *2017 australasian universities power engineering conference (aupec)* (pp. 1–6).

Rahimiyan, M., & Mashhadi, H. R. (2010). An adaptive q-learning algorithm developed for agent-based computational modeling of electricity market. *IEEE Transactions on Systems, Man, and Cybernetics Part C, Applications and Reviews*, *40*(5), 547.

Riedmiller, M. (2005). Neural fitted q iteration–first experiences with a data efficient neural reinforcement learning method. , 317–328.

Ripley, B. D. (2007). *Pattern recognition and neural networks*. Cambridge university press.

Rocchetta, R., Bellani, L., Compare, M., Zio, E., & Patelli, E. (2019). A reinforcement learning framework for optimal operation and maintenance of power grids. *Applied Energy*, *241*, 291–301.

Rocchetta, R., Compare, M., Patelli, E., & Zio, E. (2018). A reinforcement learning framework for optimisation of power grid operations and maintenance. In *8th international workshop on reliable engineering computing: computing with confidence.*

Rodrigues, L., Yoneyama, T., & Nascimento, C. (2012). How aircraft operators can benefit from phm techniques. *IEEE Aerospace Conference Proceedings*. doi: 10.1109/AERO.2012.6187376

Sankararaman, S. (2015). Significance, interpretation, and quantification of uncertainty in prognostics and remaining useful life prediction. *Mechanical Systems and Signal Processing*, *52-53*(1), 228-247. doi: 10.1016/j.ymssp.2014.05.029

Sankararaman, S., Ling, Y., Shantz, C., & Mahadevan, S. (2011). Uncertainty quantification in fatigue crack growth prognosis. *International Journal of Prognostics and Health Management*, *2*(1).

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2010a). Evaluating prognostics performance for algorithms incorporating uncertainty estimates. *IEEE Aerospace Conference Proceedings*. doi: 10.1109/AERO.2010.5446828

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2010b). Metrics for offline evaluation of prognostic performance. *International Journal of Prognostics and Health Management*, *1*(1).

Sigaud, O., & Buffet, O. (2013). *Markov decision processes in artificial intelligence*. John Wiley & Sons.

Silver, D., Huang, A., Maddison, C., Guez, A., Sifre, L., Van Den Driessche, G., ... Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, *529*(7587), 484-489. doi: 10.1038/nature16961

Simes, J., Gomes, C., & Yasin, M. (2011). A literature review

of maintenance performance measurement: A conceptual framework and directions for future research. *Journal of Quality in Maintenance Engineering*, *17*(2), 116-137. doi: 10.1108/13552511111134565

Speijker, L., Van Noortwijk, J., Kok, M., & Cooke, R. (2000). Optimal maintenance decisions for dikes. *Probability in the Engineering and Informational Sciences*, *14*(1), 101-121.

Sutton, R., Precup, D., & Singh, S. (1999). Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, *112*(1), 181-211. doi: 10.1016/S0004-3702(99)00052-1

Sutton, R. S., & Barto, A. G. (1998). *Introduction to reinforcement learning* (Vol. 135). MIT Press Cambridge.

Szepesvári, C. (2010). Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning*, *4*(1), 1–103.

Tsitsiklis, J. N., & Van Roy, B. (2002). On average versus discounted reward temporal-difference learning. *Machine Learning*, *49*(2-3), 179–191.

Van Horenbeek, A., & Pintelon, L. (2013). A prognostic maintenance policy-effect on component lifetimes. In *2013 proceedings annual reliability and maintainability symposium (rams)* (pp. 1–6).

van Noortwijk, J. (2009). A survey of the application of gamma processes in maintenance. *Reliability Engineering & System Safety*, *94*(1), 2 - 21. (Maintenance Modeling and Application) doi: https://doi.org/10.1016/j.ress.2007.03.019

Walraven, E., Spaan, M. T., & Bakker, B. (2016). Traffic flow optimization: A reinforcement learning approach. *Engineering Applications of Artificial Intelligence*, *52*, 203–212.

Wang, Y.-C., & Usher, J. M. (2005). Application of reinforcement learning for agent-based production scheduling. *Engineering Applications of Artificial Intelligence*, *18*(1), 73–82.

Zio, E. (2012). Prognostics and health management of industrial equipment. *Diagnostics and Prognostics of Engineering Systems: Methods and Techniques*, 333-356. doi: 10.4018/978-1-4666-2095-7.ch017

Zio, E. (2016). Some challenges and opportunities in reliability engineering. *IEEE Transactions on Reliability*, *65*(4), 1769-1782. doi: 10.1109/TR.2016.2591504

Zio, E., & Compare, M. (2013). Evaluating maintenance policies by quantitative modeling and analysis. *Reliability Engineering & System Safety*, *109*, 53–65.

## APPENDIX 1

To give a proper off-line estimation of the component RUL $\mathbf{P}_k$ under the different operating levels $l \in \Lambda$, we divide the system degradation level $\in [\xi, \chi]$ in $\Theta$ intervals $[\theta_0, \theta_1), \ldots, [\theta_{\Theta-1}, \theta_\Theta)$, such that $\theta_0 = \xi$, $\theta_\Theta = \chi$. Then, we assume that the RUL of the component with crack length $x_k \in [\theta_i, \theta_{i+1})$ is the same as the RUL of the component with crack length $\theta_i$, i.e., $\alpha_l(x) = \alpha_l(\theta_i)$ and $\beta_l(x) = \beta_l(\theta_i)$ $\forall x \in [\theta_i, \theta_{i+1})$.
Then, for each load level $l$ and initial crack length $x_0$ in $\theta_0, \ldots, \theta_{\Theta-1}$, we sample $S$ times the measured crack length $y_0$ and the corresponding RUL prediction $\mathbf{P}_0$. Finally, for each load level and initial crack length, we fit a Weibull distribution using the maximum likelihood method to get the values of $\alpha_l(\theta), \beta_l(\theta), \theta \in \{\theta_0, \ldots, \theta_{\Theta-1}\}, l \in \Lambda$.

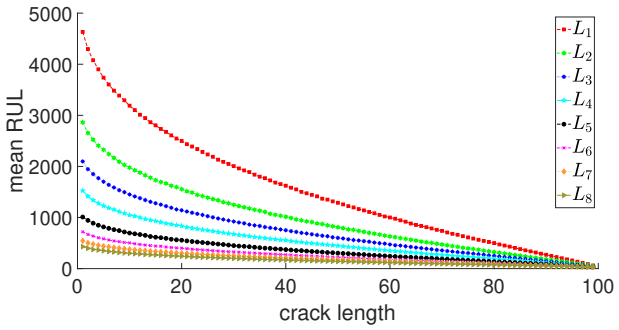


Figure 6. Average RUL value $E[\mathbf{P}_k]$ vs crack length for $L = 8$ operating levels.

**Algorithm 1** Off-line estimation of the component RUL.

**for** *load level* $l = 1, \ldots, L$ **do**
  **for** $\theta = \theta_0, ..., \theta_{\Theta-1}$ **do**
    $s = 1$
    $Predictions = \emptyset$
    **while** $s < S$ **do**
      Sample $y \sim \mathcal{N}(\theta, \sigma_y)$
      Compute the predicted RUL $P_0$ corresponding to crack length $y$ at operating level $l$ using the selected PHM algorithm
      Append $P_0$ to $Predictions$
      $s = s + 1$
    **end**
    Find $\alpha_l(\theta), \beta_l(\theta)$ fitting a Weibull distribution on $Predictions$
  **end**
**end**

The results of the described procedure using $\Theta = 99$ intervals (the same as in the case study) are summarized in Figure 6, which shows the average RUL $E[\mathbf{P}_k]$ versus crack length, for the $L = 8$ different operating levels. Figure 7 shows some of the Weibull distributions obtained using the described procedure.

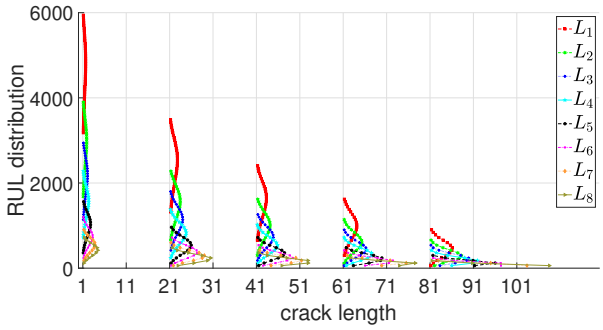


Figure 7. Estimated RUL distribution at different crack lengths for $L = 8$ operating levels.

## APPENDIX 2

**Algorithm 2** The QL+ANN Algorithm.

Set $ei = 1$, $n_{ei}$ $N_{ei}$, $\tau_\epsilon^{ei}$, $N_\alpha$, $\epsilon_0$, $\alpha_0$;

**Phase 1: Off-Line Training**

Initialize Networks $\mathcal{N}_l$ and $t = 1$, $l = 1, ..., L + 2$ with random weights;

**while** $t < n_{ei}$ **do**
  Sample transitions $\mathbf{S}_t, \mathbf{a}_t \rightarrow \mathbf{S}_{t+1}, \mathbf{a}_{t+1}$ and observe rewards $R_t$ (according to random policy);
**end**

Approximate $Q$ by the MC estimate $Y_t = \sum_{t'=t}^{t+\Phi} \gamma^{t'-t} \cdot R_{t'}$

Train each $\mathcal{N}_l$ using $\{\mathbf{S}_t | t = 1, ..., n_{ei}, \mathbf{a}_t = l\}$ and the estimated $Y_t$ (output);

**Phase 2: Learning**

**while** $ei < N_{ei}$ *(Episodic Loop)* **do**
  Set $t = 1$ Initialize state $\mathbf{S}_t$ randomly
  $\epsilon = \epsilon_0 \cdot \tau_\epsilon^{ei}$
  $\alpha = \alpha_0 \cdot \frac{N_\alpha + 1}{N_\alpha + ei}$
  **while** $t < W$ *(episode run)* **do**
    Sample $\rho$ from uniform distribution in $[0, 1]$
    **if** $\rho < 1 - \epsilon$ **then**
      $\mathbf{a}_t = \arg \max_{l \in \{1,...,L+2\}} \hat{q}_l(\mathbf{S}_t | \boldsymbol{\mu}_l)$
    **else**
      Select $\mathbf{a}_t$ randomly from all actions
    **end**
    Take action $\mathbf{a}_t$, observe $\mathbf{S}_{t+1}$ and reward $R_t$
    **Update** network $\mathcal{N}_{\mathbf{a}_t}$ weights
    $\boldsymbol{\mu}_{\mathbf{a}_t} \leftarrow \boldsymbol{\mu}_{\mathbf{a}_t} + \alpha \cdot [R_t + \gamma \cdot \max_{l \in \{1,...,|\mathcal{A}|\}} \hat{q}_l(\mathbf{S}_{t+1} | \boldsymbol{\mu}_l) - \hat{q}_{\mathbf{a}_t}(\mathbf{S}_t | \boldsymbol{\mu}_{\mathbf{a}_t})] \cdot \nabla \hat{q}_{\mathbf{a}_t}(\mathbf{S}_t | \boldsymbol{\mu}_{\mathbf{a}_t})$
    Set $t = t + 1$
  **end**
  Go to next episode $ei = ei + 1$
**end**