Proceedings of the 35th European Safety and Reliability & the 33rd Society for Risk Analysis Europe Conference
Edited by Eirik Bjorheim Abrahamsen, Terje Aven, Frederic Bouder, Roger Flage, Marja Ylönen
 Published by Research Publishing, Singapore.
doi: 10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P9167-cd

Offline Learning of Maintenance Policies Using Reinforcement Learning and Historical Maintenance Data

Quang Khai TRAN, Khac Tuan HUYNH, Antoine GRALL, Yves LANGERON

Computer Science and Digital Society Laboratory (LIST3N), University of Technology of Troyes, France.
E-mail: {quang_khai.tran, tuan.huynh, antoine.grall, yves.langeron}@utt.fr

In condition-based maintenance optimization, it is often assumed that the degradation process model is known, so that classical paradigms, such as (Markov)-renewal theory or dynamic programming, can be adopted to find this optimal policy. When degradation modeling becomes challenging, it is possible to learn such a policy directly from maintenance data. Considering offline datasets, consisting of pre- and post-maintenance system states, actions taken and associated costs, generated by various non-optimal behavior policies, our goal is to explore reinforcement learning approach to extract better maintenance policies, without any further system condition monitoring information. In the literature, offline reinforcement learning methods have been studied for maintenance optimization with discounted reward metric and discrete degradation state space, but still received less attention when considering continuous state space in the infinite horizon under the average reward metric. In this paper, we adapted a relative Q-learning algorithm with function approximation to offline settings under the average reward metric and combined it with data augmentation to learn higher performance policies from several maintenance datasets collected from continuously degrading maintained systems. Numerous results under different data configurations show that a near-optimal policy can be learned with relatively little data.

Keywords: condition-based maintenance, continuous state space, average reward, Markov decision process, reinforcement learning, relative Q-learning, function approximation, data augmentation, maintenance data

1. Introduction

Condition-based maintenance (CBM) is a class of policies that rely on the degradation states of the system to decide the maintenance activities. In existing comparative studies, CBM has been shown to provide better performance than the traditional time-based maintenance. Developing a CBM policy involves addressing two key issues: (*i*) *when to intervene on the system* and (*ii*) *what to do at that intervention time*, so that the maintenance performance is optimized.

The parametric approach is a common way to develop CBM policies whose predefined structure is parameterized by some decision variables like inspection rate, preventive maintenance or inventory level thresholds (Zhang and Zeng, 2017). Optimizing such structure returns to determine the values of decision variables that optimize certain performance metric. This approach usually relies on (Markov)-renewal theory and can handle both discrete and continuous degradation phenomena. Normally, the model of these degradation phenomena is assumed to be completely known (Huynh, 2020) or can be updated on-the-fly (Mosayebi Omshi et al., 2020) as the system operates. The main challenges of this parametric approach reside in the accuracy of degradation modeling and the appropriate choice of a predefined maintenance structure for a given system.

Another popular approach to develop CBM policies is to formulate the maintenance problem as (semi)-Markov decision processes (MDPs) (Puterman, 2005) to find the optimal policy. Unlike the parametric approach, MDP-based methods typically do not assume any structure on the policy, but search for a state-action mapping that optimizes a certain performance metric. MDP-based methods have proven their efficiency, especially for CBM problems where the state spaces are discrete (Cheng and Zhao, 2023). An MDP can be solved using dynamic programming methods (Bellman, 1984), which require a transition model of the system degradation. When degradation modeling is challenging, the optimal pol-

icy can also be learned directly from data collected from the system using reinforcement learning (RL) (Sutton and Barto, 2020). Indeed, RL is a data-driven approach to guide a maintenance decision maker (i.e., an agent) to learn an optimal policy for a specific degrading system (environment) through interactions (see Fig. 1). Specifically, at a decision epoch t, the agent will observe the current environment state s_t and take an action a_t based on this observation. The environment will then respond with a new state s_{t+1} and a reward (cost) signal r_{t+1}. Using the tuples $(s_t, a_t, s_{t+1}, r_{t+1})$, the agent seeks a policy that optimizes a performance metric, such as the expected long-term total discounted cost or the long-term average cost. This framework has been successfully applied to develop CBM policies across various sectors, including manufacturing (Paraschos et al., 2020), aircraft maintenance (Hu et al., 2021), and fuel cells management (Zuo et al., 2025). Working without a transition model, RL usually requires a substantial amount of data through online interaction with the environment. The aim of this interaction process is twofold. On the one hand, it allows the agent to explore uncharted regions of the state-action space, looking for better state-action pairs. On the other hand, it allows the agent to self-correct its policy based on newly acquired data (Rezaeifar et al., 2022). However, in many maintenance settings, such intensive exploration is not only time-consuming and costly, but also causes additional damage to the considered system. Therefore, as illustrated in Fig. 2, it is crucial to learn an efficient maintenance policy without any further online interaction with the environment (i.e., using only historical offline data). Offline RL is a promising approach for this goal.

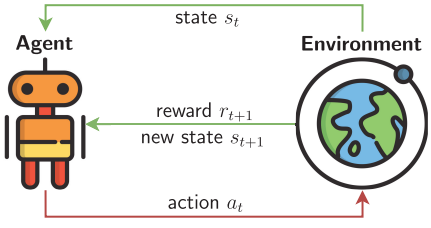

Fig. 1. Typical reinforcement learning framework.

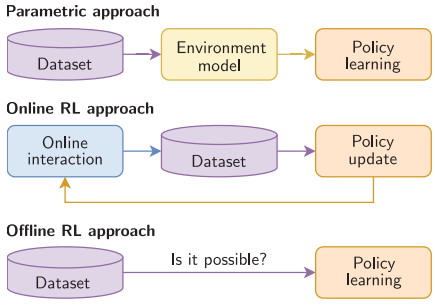

Fig. 2. Main approaches for policy learning.

In offline RL, the agent solely learns from a fixed dataset, hence does not require real-time interactions with the environment. This dataset may be collected from expert advice, maintenance activities issued by suboptimal behavior policies, or both. The goal is to develop a maintenance policy that outperforms these behavior policies, in line with the principles of off-policy learning (Sutton and Barto, 2020, Section 5.4). Two main methods are usually adopted to achieve such a maintenance policy in offline RL: *value-based* and *policy constraint* (Levine et al., 2020). The former learns the value function in a pessimistic way to reduce the bias estimation of the value function due to missing data points. Meanwhile, the latter directly constrains the policy to not differ significantly from the behavioral policy.

In the CBM literature, offline RL has received less attention compared to its online counterpart. Moreover, it has been mostly studied for maintenance optimization with discounted reward metric and discrete degradation state space. Considering continuous state space in the infinite horizon under the average reward metric, this paper focuses more specifically on learning maintenance policies from limited historical data in offline settings (see e.g., maintenance data of cylinder liners of marine engines (Giorgio et al., 2015)). To this end, we adapt the relative Q-learning algorithm with function approximation developed in (Yang et al., 2024) to offline settings under the average reward

metric, combining it with data augmentation to learn higher performance policies. We conduct the proposed method with data collected from continuously degrading systems maintained by different behavior maintenance policies: random policy, mixture of multiple suboptimal policies and one suboptimal policy. The results confirm that a near-optimal policy can be learned with relatively limited data.

The remainder of this paper is organized as follows. Section 2 introduces an improved average-reward relative Q-learning method with function approximation for continuous state space in offline settings. Section 3 describes the maintained system of interest and shows how to collect the data. We experiment with the proposed method in Section 4. Finally, some conclusions and perspectives are discussed in Section 5.

2. Approximated relative Q-learning

In this section, we first introduce the basic concepts of Q-learning under the average-reward metric, then present the relative Q-learning for discrete and continuous state spaces, and finally suggest an adapted version for offline settings.

Let us denote $\mathcal{M}$ as an MDP defined by a tuple $(\mathcal{S}, \mathcal{A}, r, p)$, where

- $\mathcal{S}$ is the state space representing all possible degradation states s of the system,
- $\mathcal{A}$ is the action space consisting of all possible actions that the agent can take,
- r is the immediate cost function for taking action a in state s then transitioning to a state s',
- p is the transition function defining the probability of transitioning from state s to a region $H \subset \mathcal{S}$, i.e., $p : \mathcal{S} \times \mathcal{A} \to \Pr(H)$.

A deterministic policy π is a mapping $\pi : \mathcal{S} \to \mathcal{A}$ that prescribes the action a to be taken by the agent when in state s, such that $\pi(s) = a$. Under a policy π, the long-term average cost is defined as

$$J(\pi) = \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_\pi \left[\sum_{t=0}^{T-1} r_{t+1} \mid s_0 = s\right]. \quad (1)$$

The goal of the learning agent is to find an optimal policy $\pi^\star$ that satisfies $J(\pi^\star) \leq J(\pi)$ for all policies π.

In value-based RL, the agent learns the *action value function* Q and uses this function to derive the optimal policy. Specifically, for a policy π assessed under the average cost metric, given the current state s and the current action a, the (relative) action value function Q measures the expected total differences between the actual cost and the average cost $J(\pi)$

$$Q_\pi(s,a) = \lim_{T\to\infty} \mathbb{E}_\pi \left[\sum_{t}^{T-1} (r_{t+1} - J(\pi)) \mid s, a\right]. \quad (2)$$

The Q function satisfies the *Bellman equation* (Bellman, 1984)

$$Q_\pi(s,a) = \bar{r}(s,a) - J(\pi) + \mathbb{E}_{s'\sim p(\cdot|s,a)}[Q_\pi(s', \pi(s'))], \quad (3)$$

where $\bar{r}(s,a) = \mathbb{E}[r(s,a,s') \mid s,a]$ stands for the expected cost of taking action a in state s. For the optimal policy $\pi^\star$, the optimal action value function, denoted as $Q^\star$, satisfies the following *Bellman optimality equation*

$$Q^\star(s,a) = \bar{r}(s,a) - J(\pi^\star) + \mathbb{E}_{s'\sim p(\cdot|s,a)}[\min_{a'} Q^\star(s', a')]. \quad (4)$$

Subsequently the optimal policy can be derived as $\pi^\star(s) = \arg\min_a Q^\star(s,a)$. Defining the Bellman operator as

$$\mathcal{B}(Q)(s,a) = \bar{r}(s,a) + \min_{a'} \left\{\mathbb{E}_{s'\sim p(\cdot|s,a)}[Q(s', a')]\right\}, \quad (5)$$

then we can approximate the value of $Q^\star$ by recursively applying the Bellman operator $\mathcal{B}$ in the following manner (Mahadevan, 1996)

$$Q^{(k+1)}(s,a) = \mathcal{B}(Q^{(k)})(s,a). \quad (6)$$

However, as mentioned by Puterman (2005), this Bellman operator is not numerical stable as the value of $Q(s,a)$ can diverge linearly after each iteration. To obtain a more stable approximation, White (1963) suggested regularizing the Q value at each iteration by using the relative update

$$Q^{(k+1)}(s,a) = \mathcal{B}(Q^{(k)})(s,a) - \min_b \left\{Q^{(k)}(s^{\text{ref}}, b)\right\} \quad (7)$$

where s^{ref} is an arbitrary reference state.

In RL settings, the transition models are typically unknown. Consequently, to estimate the Q functions, the expectation in the Bellman operator (5) is replaced by samples collected from the interaction with the environment. This approach leads to the formulation of the widely-used Q-learning update (Sutton and Barto, 2020, Chapter 6)

$$Q_{\text{new}}^{(k+1)}(s,a) = Q^{(k)}(s,a) + \alpha \cdot (Q^{(k+1)}(s,a) - Q^{(k)}(s,a)), \quad (8)$$

where α is a small scalar called the learning rate. This iterative update scheme is effective for discrete state spaces with a limited number of elements. However, it becomes impractical for large or continuous state spaces, as visiting all state-action pairs is infeasible, and some pairs may never be revisited for updates. To address this limitation, Yang et al. (2024) proposed leveraging the assumption that the Q-function is Lipschitz continuous by using the values of neighboring state-action pairs to update their Q-values. This approach enables a new update scheme where almost all visited state-action pairs are updated, and can thus serve as the foundation of the *average-reward relative Q-learning* (ARQL) algorithm for continuous state spaces. Various function approximation methods, such as support vector regression (SVR) (Smola and Schölkopf, 2004), can then be applied to the available state-action pairs to estimate the Q-function for other pairs. The readers are invited to refer to Yang et al. (2024) for the mathematical convergence analysis and the algorithm of ARQL.

The original ARQL algorithm is an off-policy algorithm developed for online learning. When the data space is not fully covered, the online interaction with the environment can help to fill the state space, making the ARQL algorithm effective. However, in many maintenance settings, such intensive interactions with the considered system are usually very costly and may result in additional system damage. To remedy this issue, augmented data can be generated using the available information from the existing dataset. Unlike the original data, these synthesized data can lead to bias in the learning process and should be regularized. Inspired by Fujimoto and Gu (2021), minimal adjustments are added into the original ARQL algorithm to enable its application in offline settings. Specifically, a so-called safety weight λ is introduced into the update formula for *augmented data*, leading to the following expression of the Q-learning update

$$\tilde{Q}_{\text{new}}^{(k+1)}(s,a) = \lambda \cdot (\tilde{Q}^{(k)}(s,a) + \alpha \cdot (\tilde{Q}^{(k+1)}(s,a) - \tilde{Q}^{(k)}(s,a))). \quad (9)$$

This hyperparameter λ reflects the degree of confidence in the augmented data, making the Q-learning update more conservative towards available original data. In practical implementations, it is typically set above 1 in minimization problems, and the higher the value of $\lambda > 1$, the more the Q values of augmented data reflect their higher risk.

3. System and maintenance dataset

We consider a single-unit system subject to continuous degradation, eventually leading to random failures. The system degradation process is modeled as a homogeneous gamma process $\mathcal{HGP}(\alpha, \beta)$ with shape parameter $\alpha > 0$ and scale parameter $\beta > 0$ (Van Noortwijk, 2009). The system fails when its degradation exceeds a non-acceptable fixed failure threshold L. The degradation and failure are assumed not self-announcing, so that periodic inspections with unit cost c_I are required to reveal the system states. Do nothing (DN) and as-good-as-new replacement are the two actions available at each inspection time. Depending on the working or failure state, the system can be replaced either preventively or correctively. Each preventive replacement (PR) costs $c_P > c_I$, while each corrective replacement (CR) costs $c_C > c_P$, and both includes already inspection cost. Besides, when the system fails, additional cost should be accounted for the downtime interval D from the failure time to the next inspection time at cost rate $c_D > 0$.

The above maintenance problem can be formulated as an MDP $\mathcal{M}$, where the elements in the tuple $(\mathcal{S}, \mathcal{A}, r, p)$ are defined as follows. The state space $\mathcal{S}$ represents the possible degradation levels of the system. The action space $\mathcal{A}$ consists

of DN, PR, and CR. The general cost function paid at time $t+1$ is defined as $r(s_t, a_t, s_{t+1}) = c(a_t) + c_D \cdot \mathbb{E}[D \mid s_t, a_t, s_{t+1}]$, where a_t denotes an action in the space $\mathcal{A}$ at time t, $c(a_t)$ is the associated cost (i.e., c_I, c_P or c_C) at time t, and $c_D \cdot \mathbb{E}[D \mid s_t, a_t, s_{t+1}]$ stands for the expected benefit loss due to potential downtime during the time interval $[t, t+1]$. The transition model p is derived by the probability computation based on the homogeneous gamma process $\mathcal{HGP}(\alpha, \beta)$. It is worth noting that the above MDP model only serves for data generation and does not require in our adapted ARQL algorithm for maintenance policy learning. In fact, the optimal policy for this continuous-state MDP has been proven to be of a control limit structure. Indeed, as illustrated in Fig. 3, the DN action is performed when the system degradation level is below a threshold $M^\star$, while the PR action is triggered when it belongs to the degradation interval $[M^\star, L)$. Please refer also to Tran et al. (2024) for more details. Once

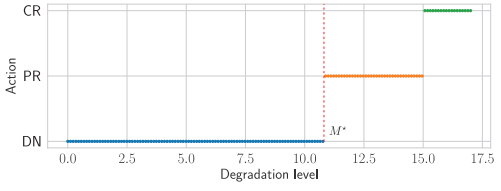

Fig. 3. Optimal policy for the CBM problem with homogeneous gamma process.

a maintenance policy μ, referred to as the behavior policy, is provided by the manufacturer, the maintenance engineer follows this policy to maintain the system and collects a dataset $\mathcal{D}$. As illustrated in Table 1, each data point in $\mathcal{D}$ is a tuple $(s_t, a_t, s_{t+1}, r_{t+1})$. Moreover, the behavior policy μ is not necessarily optimal. For instance, μ may be a control limit policy where the preventive replacement threshold M does not equal the optimal threshold $M^\star$. Additionally, the dataset $\mathcal{D}$ may combine data from multiple behavior policies. The objective is to leverage $\mathcal{D}$ to learn an improved policy that minimizes the long-term average cost.

Table 1. Samples from the dataset.

s_t	a_t	s_{t+1}	r_{t+1}
0	DN	6.5059	2
6.5059	PR	4.3533	50
⋮	⋮	⋮	⋮
16.8721	CR	2.9015	100
⋮	⋮	⋮	⋮
9.4331	DN	12.7057	2

4. Adapted ARQL implementations and numerical results

We present in this section two scenarios of dataset generation and the corresponding results of the adapted ARQL algorithm. For the case where the data coverage is not sufficient, we propose a data augmentation technique to generate additional data points for the underrepresented actions. The results demonstrate the effectiveness of the proposed method in learning the optimal policy from historical data.

For the mentioned maintenance problem, the parameters are set as follows: $\alpha = 1$, $\beta = 1$, $L = 15$, $c_I = 2$, $c_P = 50$, $c_C = 100$, and $c_D = 75$. Inspections are conducted at regular intervals of $\Delta t = 2.5$ time units. The optimal preventive replacement threshold is $M^\star \approx 10.8$ and can be found with the approximate relative value iteration algorithm (Tran et al., 2024).

4.1. *ARQL with random policy and mixed non-optimal policies*

We first address the situation where data points adequately cover the state-action space by considering two types of behavior policies μ to generate the dataset:

- Random policy: If the degradation level at inspection is below the failure threshold L, the maintenance action is selected randomly between DN and PR with equal probability. We denote the policy and this dataset as μ_R and $\mathcal{D}_{\mu_R}$, respectively.
- Control limit policies: Several control limit policies, denoted μ_M with $M \in \{8, 9, 13, 14\}$, are employed to create separate datasets of

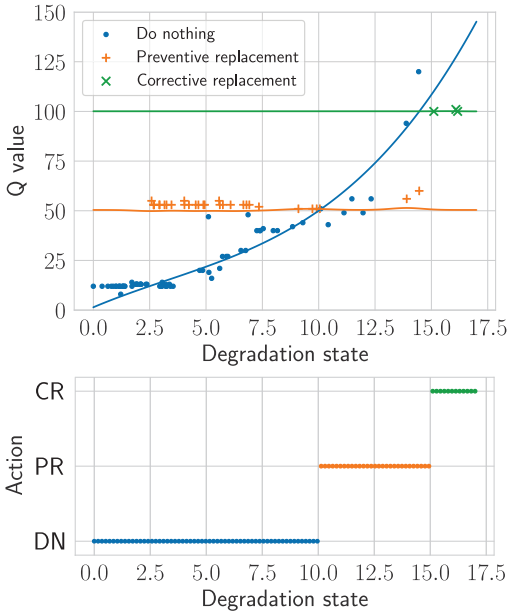

Fig. 4. Learned Q function and target policy for dataset generated by random policy.

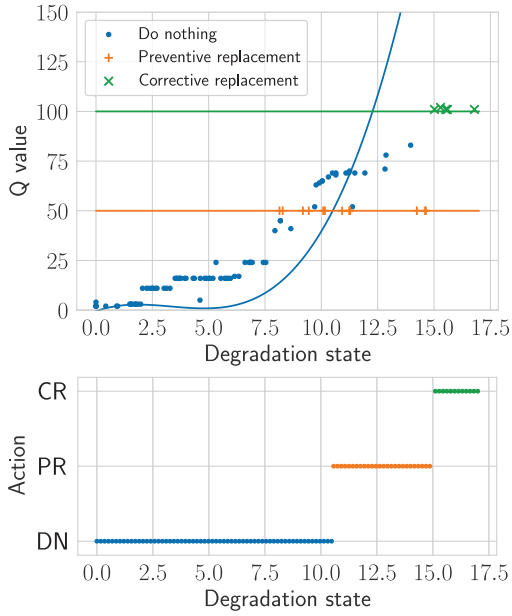

Fig. 5. Learned Q function and target policy for dataset generated by control limit policies.

equal size. These datasets are subsequently combined to form the final mixed dataset $\mathcal{D}_{\mu_M}$.

It is important to note that, except for the random policy, all other control limit policies are non-exploratory in nature. We limit the size of both $\mathcal{D}_{\mu_R}$ and $\mathcal{D}_{\mu_M}$ to 100 data points for testing purposes. This is relatively small compared to that used in existing studies employing online learning methods, which often involve hundreds of thousands of interactions. Dataset $\mathcal{D}_{\mu_R}$ and $\mathcal{D}_{\mu_M}$ are then used to train the ARQL algorithm to learn the target Q functions. Since all Q-functions in this maintenance problem can be shown to be non-decreasing, this study employs SVR with a polynomial kernel of degree 3 as the function approximation method.

Figure 4 and Figure 5 depict the target Q-functions and corresponding target policies for $\mathcal{D}_{\mu_R}$ and $\mathcal{D}_{\mu_M}$, respectively. In these figures, three distinct Q-function paths represent the actions DN, PR, and CR. For any state s, the Q-values, $Q(s,\cdot)$, can be compared, with the action associated with the lowest Q-value identified as optimal. The resulting optimal policy exhibits a control limit structure, consistent with the findings of dynamic programming. The preventive replacement thresholds for these datasets are $M_{\mu_R} = 10.1$ and $M_{\mu_M} = 10.3$, both of which are close approximations of the optimal threshold $M^\star$. These results demonstrate the effectiveness of the ARQL algorithm in learning the optimal policy.

4.2. *ARQL with single non-optimal policy and data augmentation*

For datasets generated using a single control limit behavior policy, particularly when the preventive replacement threshold $M < M^\star$, the learning algorithm may fail to approximate $Q(\cdot, \mathrm{DN})$. This occurs because no data points exist for the DN action when the degradation level exceeds M. Figure 6 illustrates the learned Q-function for a dataset generated using a single control limit policy with $M = 5$. The algorithm is unable to learn $Q(\cdot, \mathrm{DN})$ due to the overly conservative PR threshold, which prevents the collection of data for DN and CR actions. As a result, the learning agent cannot reliably decide whether DN is safe or if PR should be deployed in the region $[M, L)$. To address this issue, we propose a data augmen-

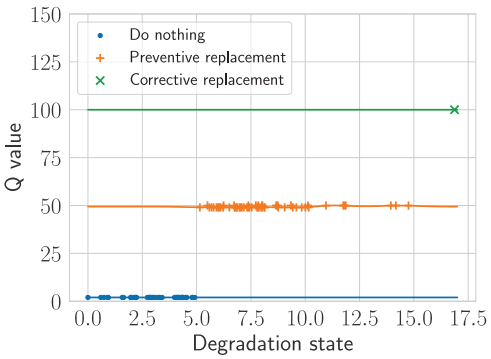

Fig. 6. Learned Q function for dataset generated using a single control limit policy ($M = 5$).

tation technique to generate additional DN data points within this region. Exploiting the properties of the $\mathcal{HGP}$—a jump process with independent increments between consecutive time steps—we define the set

$$\Xi = \left\{ \delta \mid \delta = \begin{cases} s_{t+1} - s_t & \text{if } a_t = \text{DN}, \\ s_{t+1} & \text{if } a_t = \text{PR or CR} \end{cases} \right\}$$

which contains all possible increments derived from tuples (s_t, a_t, s_{t+1}) in $\mathcal{D}$. For each pair (s_t, PR) in the dataset, a "virtual" jump of size $\xi \sim \Xi$ can be sampled to generate an augmented data point $(\tilde{s}_t, \text{DN}, \tilde{s}_{t+1}, \tilde{r}_{t+1})$, where $\tilde{s}_t = s_t$ and $\tilde{s}_{t+1} = \tilde{s}_t + \xi$. This process is repeated iteratively to populate the desired state space region. The augmented data point must also include the associated cost $\tilde{r}_{t+1}$ for the DN action. Calculating the expected downtime cost when $\tilde{s}_{t+1}$ exceeds the failure threshold L without knowing the MDP dynamics can be challenging. However, given the constant degradation rate of the $\mathcal{HGP}$, we can assume a linear degradation trajectory between $\tilde{s}_t$ and $\tilde{s}_{t+1}$, the expected downtime can be approximated using elementary geometric principles (Fig. 7). When using augmented data in the ARQL algorithm, the Q-values of the augmented data points are assigned higher safety weights λ to account for their synthetic nature. We generate 25% additional data points (equivalent to 25 data points) for the DN action. Figure 8 illustrates the learned Q-function for the augmented dataset. The target policy derived from the augmented dataset

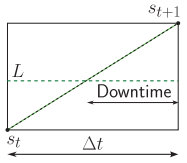

Fig. 7. Computing the expected downtime for the augmented data point.

yields a PR threshold of $M_{\mu_5} = 8$, representing a significant improvement over the original behavior policy.

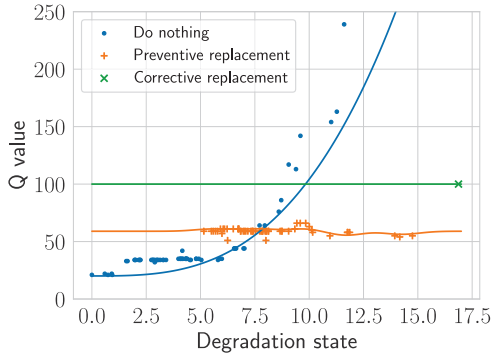

Fig. 8. Learned Q function with augmented data points.

Table 2 presents the long-run average cost rates for different policies. The conservative policy incurs a significantly higher cost than the optimal $M^\star$, while the learned policies from the random and mixed datasets perform near-optimally with only slight cost increases. The augmented dataset policy improves upon the conservative one but remains suboptimal. These results confirm the effectiveness of ARQL and data augmentation in learning near-optimal policies.

Table 2. Comparison of long-run average costs $J(\cdot)$ for different preventive replacement thresholds.

$M^\star$	M	M_{μ_R}	M_{μ_M}	M_{μ_5}
5.08	7.92 ↑ 55.91%	5.18 ↑ 1.97%	5.16 ↑ 1.57%	5.79 ↑ 14.00%

5. Conclusion and perspectives

In this paper, we have presented an offline RL approach to learn maintenance policies from historical data. Results show that even with limited data, if the dataset adequately covers the state-action space and the action space is manageable, near-optimal policies can be learned. The proposed data augmentation technique can be used to generate additional data points for actions that are underrepresented in the dataset. Future work will focus on extending the proposed method to more complex systems with multiple components and heterogeneous degradation rates.

Acknowledgement

This work is funded by the Grand Est region and the French Ministry of Higher Education and Research.

References

Bellman, R. (1984). *Dynamic Programming*. Princeton, NJ: Princeton Univ. Pr.

Cheng, W. and X. Zhao (2023, December). Maintenance optimization for dependent two-component degrading systems subject to imperfect repair. *Reliability Engineering & System Safety 240*, 109581.

Fujimoto, S. and S. S. Gu (2021). A minimalist approach to offline reinforcement learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan (Eds.), *Advances in Neural Information Processing Systems*, Volume 34, pp. 20132–20145. Curran Associates, Inc.

Giorgio, M., M. Guida, and G. Pulcini (2015, May). A condition-based maintenance policy for deteriorating units. an application to the cylinder liners of marine engine. *Applied Stochastic Models in Business and Industry 31*(3), 339–348.

Hu, Y., X. Miao, J. Zhang, J. Liu, and E. Pan (2021, March). Reinforcement learning-driven maintenance strategy: A novel solution for long-term aircraft maintenance decision optimization. *Computers & Industrial Engineering 153*, 107056.

Huynh, K. (2020, January). Modeling past-dependent partial repairs for condition-based maintenance of continuously deteriorating systems. *European Journal of Operational Research 280*(1), 152–163.

Levine, S., A. Kumar, G. Tucker, and J. Fu (2020, November). Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems.

Mahadevan, S. (1996). Average reward reinforcement learning: Foundations, algorithms, and empirical results. *Machine Learning 22*(1/2/3), 159–195.

Mosayebi Omshi, E., A. Grall, and S. Shemehsavar (2020, April). A dynamic auto-adaptive predictive maintenance policy for degradation with unknown parameters. *European Journal of Operational Research 282*(1), 81–92.

Paraschos, P. D., G. K. Koulinas, and D. E. Koulouriotis (2020, July). Reinforcement learning for combined production-maintenance and quality control of a manufacturing system with deterioration failures. *Journal of Manufacturing Systems 56*, 470–483.

Puterman, M. L. (2005). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Hoboken, NJ: Wiley-Interscience.

Rezaeifar, S., R. Dadashi, N. Vieillard, L. Hussenot, O. Bachem, O. Pietquin, and M. Geist (2022, June). Offline reinforcement learning as anti-exploration. *Proceedings of the AAAI Conference on Artificial Intelligence 36*(7), 8106–8114.

Smola, A. J. and B. Schölkopf (2004, August). A tutorial on support vector regression. *Statistics and Computing 14*(3), 199–222.

Sutton, R. S. and A. Barto (2020). *Reinforcement Learning: An Introduction* (Second edition ed.). Adaptive Computation and Machine Learning. Cambridge, Massachusetts London, England: The MIT Press.

Tran, Q. K., K. T. Huynh, A. Grall, Y. Langeron, and E. Mosayebi Omshi (2024). Average reward reinforcement learning for optimizing condition based maintenance policies of continuously deteriorating systems. In *Advances in Reliability, Safety and Security*, ESREL 2024 Monograph Book Series, pp. 213–222. Polish Safety and Reliability Association.

Van Noortwijk, J. (2009, January). A survey of the application of gamma processes in maintenance. *Reliability Engineering & System Safety 94*(1), 2–21.

White, D. (1963, June). Dynamic programming, markov chains, and the method of successive approximations. *Journal of Mathematical Analysis and Applications 6*(3), 373–376.

Yang, X., J. Hu, and J.-Q. Hu (2024). Relative Q-learning for Average-Reward Markov Decision Processes with Continuous States. *IEEE Transactions on Automatic Control*, 1–14.

Zhang, X. and J. Zeng (2017, October). Joint optimization of condition-based opportunistic maintenance and spare parts provisioning policy in multiunit systems. *European Journal of Operational Research 262*(2), 479–498.

Zuo, J., N. Y. Steiner, Z. Li, C. Cadet, C. Bérenguer, and D. Hissel (2025, April). Reinforcement learning-based maintenance scheduling for a stochastic deteriorating fuel cell considering stack-to-stack heterogeneity. *Reliability Engineering & System Safety 256*, 110700.