Advances in Reliability, Safety and Security, Part 4 – Kołowrocki, Dąbrowska (eds.)
 ISBN 978-83-68136-16-6 (printed), ISBN 978-83-68136-03-6 (electronic)

Advances in Reliability, Safety and Security

ESREL 2024
Monograph Book Series

Average Reward Reinforcement Learning For Optimizing Condition Based Maintenance Policies Of Continuously Deteriorating Systems

Quang Khai Tran, Khac Tuan Huynh, Antoine Grall, Yves Langeron, Elham Mosayebi Omshi

University of Technology of Troyes, Troyes, France

Abstract

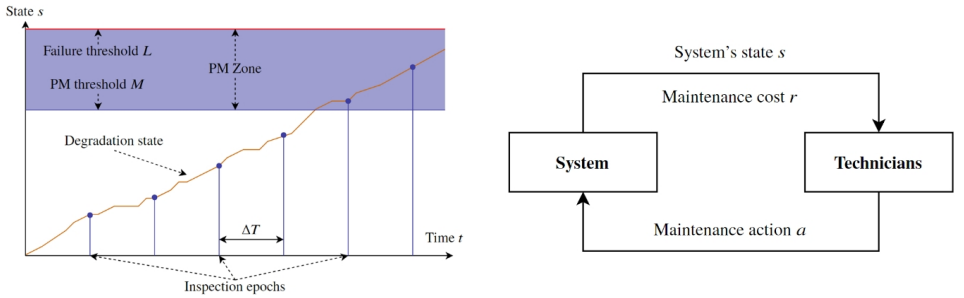
This paper investigates the optimization of condition-based maintenance policies for systems experiencing continuous degradation. We address this challenge by formulating it within the framework of Markov decision process and employing reinforcement learning techniques to derive optimal policies based on the long-run total maintenance cost per time unit. In contrast to existing approaches that often discretize the continuous state space to facilitate the application of reinforcement learning algorithms, we directly handle the continuous state space in both problem statement and solution method, thus preserving its Markov property. The reward function is also carefully designed to describe real-world situations by incorporating the downtime cost rate. Additionally, we treat the maintenance problem as a continual task rather than an episodic one, enabling us to identify optimal policies that minimize the long-run total maintenance cost per time unit. The obtained maintenance policies exhibited a strong correspondence with the influences of the degradation process. Furthermore, when compared to the discretized state space, it demonstrated better performances.

Keywords: condition-based maintenance, reinforcement learning, Markov decision process, degradation process, continuous state space

1. Introduction

Condition-based maintenance (CBM) is a significant category in the field of maintenance, involving inspections conducted on a system to gather operational data, including factors like reliability and degradation (Alaswad et al., 2017). Based on this information, technicians can make informed decisions about whether maintenance should be performed. The main objectives of CBM are to determine the optimal timing for inspections and the specific condition level at which maintenance actions should be implemented, effectively managing and reducing overall maintenance costs. There are two formulations for a CBM problem: ***parametric, structured problem***, and ***non-parametric, sequential decision-making problem*** (see Figure 1). The former has been extensively studied, with a parameterized policy structure optimized to achieve performance metrics like long-run cost rate. A typical topic in this approach is the control limit policy, which is parameterized by replacement thresholds (Mosayebi Omshi et al., 2021) and ordering thresholds (Wang et al., 2008). This approach offers analytical solutions and a modular property that allows modification of initial conditions while retaining the same problem structure. However, it has disadvantages, including the assumption that the underlying process describing the system's condition is initially known, which can be challenging for complex systems with no suitable models. Additionally, the predefined structure of the maintenance policy may not be the best fit for the system. In contrast, the latter approach does not assume any system structures and defines the system elements and behaviors naturally. The optimal maintenance policy is determined for each state of the system, optimizing performance metrics. One advantage of this approach is the ability to learn the optimal policy through experience, using reinforcement learning (Sutton and Barto, 2020). However, a significant drawback is the requirement for a large amount of data to learn the optimal policy, even when the condition model is known.

Furthermore, this approach does not scale well when the system's state space is large or continuous. To overcome these challenges, some works on this direction discretized the state space (Zhang et al., 2016; Chen et al., 2015), or directly assumed that the state space is finite by nature (Chen et al., 2005). These studies employed a naïve (uniformly) discretization approach to the state space, which unfortunately fails to guarantee the preservation of the Markovian property within the state-action spaces. It is also worth noting that most works use the *discounted rewards metric* to solve the optimization problem. However, the discounted setting has been proved to find the sub-optimal policies when the discount factor is poorly chosen, especially for continual tasks (Tadepalli and Ok, 1998). In the context of the *average reward metric*, only a small number of studies in the CBM field have employed this approach (Peng and Feng., 2021; Adsule et al., 2020; Xanthopoulos et al., 2018; Ling et al., 2018) compared to the prevalent use of the discounted setting. The algorithms utilized to solve the MDP in these studies can be classified into three categories: R-learning (Schwartz, 1993), SMART (Das et al., 1999), and value iteration. Among these works, Peng and Feng (2021) were the only ones to consider a continuous state space. They employed Gaussian processes (Rasmussen and Williams, 2008) to model the degradation process and approximate the value function. However, the value iteration algorithm used in their study, which relied on the original Bellman optimality equation for average reward, may not converge to an optimal solution (Mahadevan, 1996; Gosavi, 2013). To address this issue, the relative value iteration algorithm (White, 1963) can be utilized. The relative value iteration algorithm serves as the foundation for the SMART class algorithm.



(a) Parametric approach

(b) Sequential decision-making approach

Fig. 1. Example of parametric problem and sequential decision-making problem for CBM. The preventive maintenance threshold (PM) and the inter-inspection interval Δt are the parameters of the parametric problem.
In sequential decision-making problem, we find a mapping from state to action so that the performance metric is optimized.

This paper formulates the CBM problem as a sequential decision-making problem without assuming any policy structure. The problem is described as a Markov decision process (MDP), a powerful mathematical framework for modeling sequential decision-making problems and solving them under the long-term average reward metric. Traditional dynamic programming methods like value iteration (VI) or policy iteration (PI) are applicable only to finite state-action spaces. Hence, many studies attempt to discretize the state space and utilize VI and PI as aforementioned. However, this approach sacrifices the Markovian property of state space (Wiering et al., 2012). In contrast, this paper directly addresses the continuous properties of the state space while preserving its Markovian property.

2. Problem description

We investigate an offline CBM planning problem for a single-unit system that undergoes continuous degradation over time. The degradation level of the system can only be determined through periodic inspections carried out at fixed intervals of Δt time units. The degradation level can only be revealed through inspection. We also make the assumption that the failure stage is not self-announcing. If this level exceeds a predetermined threshold L, the maintenance engineer (referred to as the agent) is only allowed to perform corrective maintenance on the system. Alternatively, if the level is below this threshold, the agent must decide between conducting preventive maintenance or forgoing maintenance altogether. It is assumed that both corrective maintenance and preventive maintenance will restore the system to its original, new condition. That is, both the degradation level and the age of the system are zeros. At each inspection epoch, the agent incurs certain

maintenance costs. These costs are calculated based on the current state of the system and the most recent state-action pair. Initially, the agent must incur fixed costs according to the chosen maintenance action:

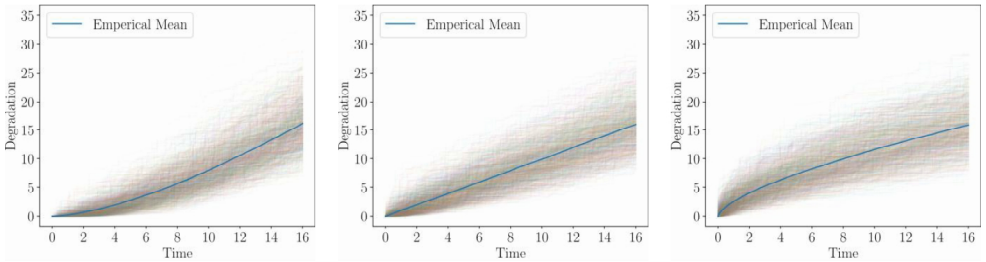
- if the agent selects *no maintenance* (NM), they are solely responsible for the inspection cost c_I;
- if the agent opts for *preventive maintenance* (PM), they must cover both the preventive maintenance cost and the inspection cost; the total cost, which encompasses both of these expenses, is represented as c_P;
- if the agent decides on *corrective maintenance* (CM), they have the responsibility of covering both the expenses related to the corrective maintenance itself and the associated inspection; these costs, when combined, are referred to as c_C.

Furthermore, depending on the current state of the system, additional costs may arise. If the degradation level exceeds the threshold L, the agent must bear downtime costs due to system failure between the current inspection and the most recent previous inspection. Conversely, if the degradation level does not exceed the threshold, no additional costs are incurred. This CBM planning issue can be perceived as a sequential decision-making problem, with the objective of identifying the optimal maintenance policy that minimizes the average cost rate of maintenance. To address this problem, we will employ the MDP framework to formulate the CBM problem and utilize the modified relative value iteration algorithm to solve the MDP and obtain the optimal solution.

In an MDP-based problem, we have an agent interacting inside an environment and being rewarded for each action it takes in this environment. Formally, a discrete-time MDP is 4-tuple $(\mathcal{S}, \mathcal{A}, r, p)$: $\mathcal{S}$ is the state space containing all possible states of the environment; $\mathcal{A}$ is the action space including all possible actions that the agent can perform in this environment; $r(\boldsymbol{s}, a, \boldsymbol{s}')$ is the immediate reward function, it is a 3-argument function that computes the reward that the agent can receive when performing action a in state $\boldsymbol{s}$, resulting the next state is $\boldsymbol{s}'$; and $p(\boldsymbol{s}, a, \boldsymbol{s}')$ is a model of the environment, given the current state and the current action, the model provides us the information about the next states that the agent can transition to. The agent interacts with the environment using a policy π, which is a mapping from state to action: $\pi(\boldsymbol{s}) = a$ (we only consider stationary deterministic policy in this problem). To solve an MDP is to find an optimal policy $\pi^\star$ that optimizes a metric of interest (total discounted rewards, average reward per action taken, etc.).

2.1. State space $\mathcal{S}$

The MDP environment in this context corresponds to the degrading system under consideration. Given that the degradation process is non-stationary, it becomes necessary to incorporate the system's age alongside the degradation level to accurately determine its current operational condition. For instance, assuming a convex degradation curve, a system with low degradation level and low age would pose minimal risk of failure in the near future. On the other hand, even if the degradation level is low, a system with high age would carry a higher risk of approaching the failure stage soon (see Figure 2). Consequently, a system's state at a time index $t_i, i \in \mathbb{N}$ can be represented as a two-element vector $S_{t_i} = [W_{t_i}, Z_{t_i}]$, where $W_{t_i} \geq 0$ denotes the degradation level at time t_i, and $Z_{t_i} \geq 0$ indicates the age of the system at time t_i. Thus, the state space is defined as $\mathbb{R}_+^2$, encompassing non-negative values in a two-dimensional Euclidean space. For a degradation level W_{t_i}, if it exceeds a pre-defined failure threshold L, we consider the system is in the failure stage.



(a) Convex degradation paths (b) Linear degradation paths (c) Concave degradation paths

Fig. 2. Different degradation paths. For each case, we simulate 1000 degradation paths, the blue solid lines represent the mean paths.

2.2. Action space $\mathcal{A}$

Within this environment, the agent has the capability to undertake three distinct maintenance actions, each represented by an integer value: no maintenance (0), preventive maintenance (1), and corrective maintenance (2).

The action space, denoted as $\mathcal{A}$, is defined as $\mathcal{A} = \{0,1,2\}$. To indicate the available actions in a given state S_{t_i}, we use the notation $\mathcal{A}(S_{t_i})$. Specifically, we assume the following mappings:

$$\mathcal{A}(S_{t_i} = [W_{t_i}, Z_{t_i}]) = \begin{cases} \mathcal{A}([0,0]) = \{0\}, \\ \mathcal{A}([W_{t_i}, Z_{t_i}], W_{t_i} \geq L, Z_{t_i} > 0) = \{2\}, \\ \mathcal{A}([W_{t_i}, Z_{t_i}], 0 \leq W_{t_i} < L, Z_{t_i} > 0) = \{0,1\}. \end{cases} \quad (1)$$

Regarding the effects of preventive maintenance and corrective maintenance, both actions will result in an immediate transition of the system to the state [0, 0] at the current time step. More formally, at time t_i, let $W_{t_i^-}$ and $Z_{t_i^-}$ represent the degradation level and age of the system before taking any maintenance action. Similarly, let $W_{t_i^+}$ and $Z_{t_i^+}$ denote the degradation level and age of the system immediately after the completion of maintenance. In these cases, the followings hold true: $W_{t_i^+} = 0$ and $Z_{t_i^+} = 0$ if corrective or preventive maintenance is implemented, and $W_{t_i^-} = W_{t_i^+} = W_{t_i}, Z_{t_i^-} = Z_{t_i^+} = Z_{t_i}$ if no maintenance is undertaken (see Figure 3 for an illustration). When mentioning S_{t_i}, W_{t_i}, or Z_{t_i} without any additional clarification, we adopt the convention that we are referring to $S_{t_i^-}, W_{t_i^-}$, and $Z_{t_i^-}$.

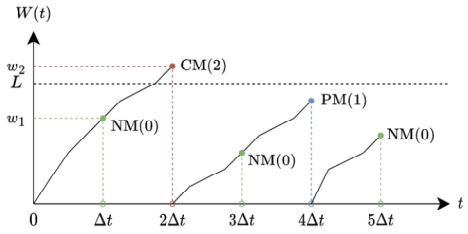


Fig. 3. Effects of maintenance actions. At decision time $t_1 = \Delta t$, after inspection, we make a decision to have no maintenance (NM (0)). Therefore, we have $W_{t_1^-} = W_{t_1^+} = w_1$ and $Z_{t_1^-} = Z_{t_1^+} = \Delta t$. At decision time $t_2 = 2\Delta t$, after inspection, we know that $W_{t_2^-} = w_2 > L$ and $Z_{t_2^-} = 2\Delta t$. Then we decide to conduct corrective maintenance (CM (2)). Thus, $W_{t_2^+} = 0$ and $Z_{t_2^+} = 0$. At $t_4 = 4\Delta t$, we conduct preventive maintenance (PM (1)). In this case, we can derive $W_{t_4^-}, W_{t_4^+}$ and $Z_{t_4^-}, Z_{t_4^+}$ in a similar manner when CM is chosen.

2.3. Model p

We start this part by introducing the gamma random variable (Van Noortwijk, 2009), which will be extensively employed in subsequent analyses. A random variable X that adheres to a gamma distribution is denoted as $X \sim \mathcal{G}(\alpha, \beta)$. The gamma distribution represents a continuous probability distribution, characterized by the subsequent probability density function:

$$f(x; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} \cdot \mathbb{1}_{x>0}, \quad (2)$$

where α and β are two positive constants called the shape and the scale parameters, respectively.

The natural degradation process is modelled by a non-stationary gamma process $(W_{t_i})_{t_i \geq 0}$ with the shape function $\Lambda(Z_{t_i})$ and the scale parameter β. A gamma process is a jump process with independent increments. At two distinct time t_1 and t_2 that satisfy $0 < t_1 < t_2$, the increment in degradation level $\Delta W = W_{t_2} - W_{t_1}$ is a gamma random variable with shape parameter $\Lambda(Z_{t_2^-}) - \Lambda(Z_{t_1^+})$ and the scale parameter β, that is $\Delta W \sim \mathcal{G}(\Lambda(Z_{t_2^-}) - \Lambda(Z_{t_1^+}), \beta)$ (Kahle et al., 2016). Remember that when preventive maintenance or corrective maintenance is conducted at time step t_i, the age and the degradation level will drop to zero at the same time step, that is $W_{t_i^+} = 0, Z_{t_i^+} = 0$ (see the action space section above). In general, $Z_{t_{i+1}^-} = Z_{t_i^+} + \Delta t$. In this problem, the model p is the probability density function (PDF) for the next degradation level $W_{t_{i+1}}$, given the current state $S_{t_i} = [W_{t_i}, Z_{t_i}]$ and the current action A_{t_i}. Regarding the next age $Z_{t_{i+1}}$ of the system, since the inter-inspection interval is fixed, we can easily compute the next age deterministically as $Z_{t_{i+1}^-} = Z_{t_i^+} + \Delta t$. If $A_{t_i} = 0$, the PDF of the next degradation $W_{t_{i+1}^-}$ is:

$$p\left(W_{t_{i+1}} = x \mid S_{t_i} = [W_{t_i}, Z_{t_i}], A_{t_i} = 0\right) = \frac{\beta^{\Lambda\left(Z_{t_i^+ + \Delta t}\right) - \Lambda\left(Z_{t_i^+}\right)}}{\Gamma\left(\Lambda\left(Z_{t_{i+1}^-}\right) - \Lambda\left(Z_{t_i^+}\right)\right)} \left(x - W_{t_i^+}\right)^{\Lambda\left(Z_{t_{i+1}^-}\right) - \Lambda\left(Z_{t_i^+}\right) - 1} e^{\beta(x - W_{t_i})} \mathbb{1}_{\left(x > W_{t_i^+} = W_{t_i}\right)}. \quad (3)$$

When $A_{t_i} = 1$ or $A_{t_i} = 2$, we have the same PDF for these two actions as their effects are the same (note that the costs for these two actions are still different):

$$p\left(W_{t_{i+1}} = x \mid S_{t_i} = [W_{t_i}, Z_{t_i}], A_{t_i} = \{1,2\}\right) = \frac{\beta^{\Lambda\left(Z_{t_i^-}+\Delta t\right)-\Lambda\left(Z_{t_i^+}\right)}}{\Gamma\left(\Lambda\left(Z_{t_{i+1}^-}\right)-\Lambda\left(Z_{t_i^+}\right)\right)} x^{\Lambda\left(Z_{t_{i+1}^-}\right)-\Lambda\left(Z_{t_i^+}\right)-1} e^{\beta(x)} \mathbb{1}_{\left(x>W_{t_i^+}=0\right)}. \quad (4)$$

2.4. Immediate reward function r

Before defining our immediate reward function, we discuss a little on the reward function. Some works consider the reward function as a 2-argument function $r(\boldsymbol{s}, a)$ (Yousefi et al., 2022; Adsule et al., 2020). This kind of reward function will be problematic if we consider the downtime cost. In this case, we can only treat the downtime cost as a constant as we do not have enough information to compute the (expected) downtime between two states $S_{t_i} = [W_{t_i}, Z_{t_i}]$ and $S_{t_{i+1}} = [W_{t_{i+1}}, Z_{t_{i+1}}]$. Thus, we define the immediate reward function in our work as a 3-argument function and prove that it is a better approach. Given two consecutive time steps t_i and t_{i+1}, the immediate reward function $r(\boldsymbol{s}, a, \boldsymbol{s}')$ computes the reward $R_{t_{i+1}}$ for performing action $A_{t_i} = a$ in state $S_{t_i} = \boldsymbol{s}$ and then transitioning to state $S_{t_{i+1}} = \boldsymbol{s}'$. In comparison to certain existing studies that assume a constant downtime cost, our approach recognizes the dynamic nature of real-world scenarios. We account for the downtime duration resulting from failures, understanding that extended periods of downtime correspond to higher financial losses. We define the following one-step reward function (recall that a state S_{t_i} is a two-component vector: $S_{t_i} = [W_{t_i}, Z_{t_i}]$):

$$r(\boldsymbol{s}, a, \boldsymbol{s}') = R_{t+1} = \begin{cases} c_I + c_D \cdot \mathbb{E}\left[D_0\left(S_{t_i}, S_{t_{i+1}}\right)\right] \cdot \mathbb{1}_{W_{t_{i+1}} \geq L}, \text{if } W_{t_i} < L, A_{t_i} = 0 \text{ (NM at } t_i), \\ c_P + c_D \cdot \mathbb{E}\left[D_1\left(S_{t_i}, S_{t_{i+1}}\right)\right] \cdot \mathbb{1}_{W_{t_{i+1}} \geq L}, \text{if } W_{t_i} < L, A_{t_i} = 1 \text{ (PM at } t_i), \\ c_C + c_D \cdot \mathbb{E}\left[D_2\left(S_{t_i}, S_{t_{i+1}}\right)\right] \cdot \mathbb{1}_{W_{t_{i+1}} \geq L}, \text{if } W_{t_i} \geq L, A_{t_i} = 2 \text{ (CM at } t_i). \end{cases} \quad (5)$$

In Eq. (5), c_I, c_P, c_C and c_D are the inspection cost, preventive maintenance cost, corrective maintenance cost, and the downtime cost rate, respectively. We utilize the notation $D_a\left(S_{t_i}, S_{t_{i+1}}\right), a \in \mathcal{A}(S_{t_i})$ to represent the duration of system downtime between time steps t_i and t_{i+1}, considering the specific action a, as well as the states S_{t_i} and $S_{t_{i+1}}$. As the failure stage is not self-announcing, the duration of downtime $D_a\left(S_{t_i}, S_{t_{i+1}}\right)$ is subject to randomness and can be considered as a random variable. Consequently, it is appropriate to utilize the concept of expectation $\mathbb{E}[D_a(S_{t_i}, S_{t_{i+1}})]$ to quantify this quantity. The expression $\mathbb{1}_{W_{t_{i+1}} \geq L}$ represents the indicator function that signifies whether the degradation level $W_{t_{i+1}}$ of the state $S_{t_{i+1}}$ surpasses the failure threshold L. If the triplet $(S_{t_i}, A_{t_i}, S_{t_{i+1}})$ is known, it is possible to calculate the expected downtime duration analytically. The process for deriving this expectation will be presented below. Additionally, we make the assumption that at time step $t_0 = 0$, the system is consistently in an as-good-as-new state denoted as $S_0 = [0,0]$, and the agent always selects action $A_0 = 0$. Furthermore, it can be seen that in the immediate reward function, we can observe that there is always a deterministic part c_I, c_P, or c_C, depending on the corresponding maintenance action. This part only depends on the current state S_{t_i} and the current action A_{t_i}, we use the notation $c\left(S_{t_i}, A_{t_i}\right)$ to denote this part.

The remaining work in this part is to compute the expected downtime $\mathbb{E}\left[D_a\left(S_{t_i}, S_{t_{i+1}}\right)\right]$. Clearly, $\mathbb{E}\left[D_a\left(S_{t_i}, S_{t_{i+1}}\right)\right] = 0$ if $W_{t_{i+1}^-} < L$. Considering the case $W_{t_{i+1}^-} \geq L$, between age $Z_{t_i^+}$ and age $Z_{t_{i+1}^-} = Z_{t_i^+} + \Delta t$, the system must be in the failure stage for some period. Let W_h be a degradation level where h satisfies $Z_{t_i^+} < h < Z_{t_{i+1}^-}$. Then $W_h = W_{t_i^+} + Y(W_{t_{i+1}^-} - W_{t_i^+})$ where $Y \sim \mathcal{B}(\alpha_h, \beta_h)$ is a beta distributed random variable with two shape parameters $\alpha_h = \Lambda(h) - \Lambda\left(Z_{t_i^+}\right)$ and $\beta_h = \Lambda\left(Z_{t_i^+}\right) - \Lambda(h)$. We can obtain the CDF of W_h as $F_{W_h}(x) = F_Y\left(\frac{x - W_{t_i^+}}{W_{t_{i+1}^-} - W_{t_i^+}}\right)$ where $F_Y(x)$ is the CDF of Y. Let $\mathbb{1}_{W_h \geq L | W_{t_i^+} < L, W_{t_{i+1}^-} \geq L}(h)$ be the indicator function for the event "system is failure stage $W_h \geq L$ at age $h, Z_{t_i^+} < h < Z_{t_{i+1}^-}$. Then the downtime between age $Z_{t_i^+}$ and age $Z_{t_{i+1}^-} = Z_{t_i^+} + \Delta t$ is $D_a\left(S_{t_i}, S_{t_{i+1}}\right) = \int_{Z_{t_i^+}}^{Z_{t_{i+1}^-}} \mathbb{1}_{W_h \geq L | W_{t_i^+} < L, W_{t_{i+1}^-} \geq L}(h)\, dh$. The expectation of $D_a\left[S_{t_i}, S_{t_{i+1}}\right]$ is

$$\mathbb{E}\left[D_a\left(S_{t_i}, S_{t_{i+1}}\right)\right] = \mathbb{E}\left[\int_{Z_{t_i^+}}^{Z_{t_{i+1}^-}} \mathbb{1}_{W_h \geq L | W_{t_i^+} < L, W_{t_{i+1}^-} \geq L}(h)\, dh\right] = \int_{Z_{t_i^+}}^{Z_{t_{i+1}^-}} (1 - F_{W_h}(L))\, dh. \quad (6)$$

Additionally, when the immediate reward function is a 3-argument function, we can straightforwardly derive that the expected reward for a state-action pair is a composition of an immediate reward for taking an action and a transitioning reward between current state and possible future states. This can be sketched as follows. Let $\mu_R(S_{t_i}, A_{t_i})$ be the expected reward that the agent can receive for the state-action pair (S_{t_i}, A_{t_i}) where $S_{t_i} = [W_{t_i}, Z_{t_i}]$. We have that:

$$\mu_R(S_{t_i}, A_{t_i}) = \mathbb{E}\left[R_{t_{i+1}} \mid S_{t_i}, A_{t_i} \right] = \int_{\mathcal{S}} r(S_{t_i}, A_{t_i}, S_{t_{i+1}}) p(S_{t_{i+1}} \mid S_{t_i}, A_{t_i}) dS_{t_{i+1}}. \tag{7}$$

Note that when A_{t_i} is chosen, we immediately know the next age $Z_{t_{i+1}}$. Thus, we only need to determine the next degradation level $W_{t_{i+1}}$. The model $p(S_{t_{i+1}} \mid S_{t_i}, A_{t_i})$ is just the probability density function for the next degradation level $W_{t_{i+1}}$. Therefore, we can abuse the notation and write:

$$\begin{aligned} \mu_R(S_{t_i}, A_{t_i}) &= \int_0^{\infty} r(S_{t_i}, A_{t_i}, S_{t_{i+1}}) p(W_{t_{i+1}} \mid S_{t_i}, A_{t_i}) dW_{t_{i+1}} \\ &= \int_0^{L} r(S_{t_i}, A_{t_i}, S_{t_{i+1}}) p(W_{t_{i+1}} \mid S_{t_i}, A_{t_i}) dW_{t_{i+1}} + \int_L^{\infty} r(S_{t_i}, A_{t_i}, S_{t_{i+1}}) p(W_{t_{i+1}} \mid S_{t_i}, A_{t_i}) dW_{t_{i+1}} \\ &= c(S_{t_i}, A_{t_i}) + c_D \int_L^{\infty} \mathbb{E}\left[D_{A_{t_i}}(S_{t_i}, S_{t_{i+1}}) \right] p(W_{t_{i+1}} \mid S_{t_i}, A_{t_i}) dW_{t_{i+1}}. \end{aligned} \tag{8}$$

The last integral in Eq. (8) represents the expected downtime between the current inspection and the next inspection epoch, given the current state-action pair. We can denote this quantity as $\mathbb{E}\left[D_{A_{t_i}}(S_{t_i}) \right]$. Instead of plugging the results of Eq. (6) to Eq. (8), we can compute $\mathbb{E}\left[D_{A_{t_i}}(S_{t_i}) \right]$ in an easier way. Let W_j be the degradation level of the system at age j. After maintenance action A_{t_i} is carried out, the current age will be $Z_{t_i}^+$ and the next age will be $Z_{t_{i+1}} = Z_{t_i}^+ + \Delta t$. Let $\mathbb{1}_{W_{j \geq L} \mid W_{t_i} = w}$ be the indicator function for the event: the degradation level W_j at age j is greater than or equal to L, given the current degradation W_{t_i} is w. Then the downtime between age $Z_{t_i}^+$ and age $(Z_{t_i}^+ + \Delta t)$, given the current state-action pair (S_{t_i}, A_{t_i}) is:

$$D_{A_{t_i}}(S_{t_i}) = \int_{Z_{t_i}^+}^{Z_{t_i}^+ + \Delta t} \mathbb{1}_{W_{j \geq L} \mid W_{t_i} = w} \, dj. \tag{9}$$

Clearly, $D_{A_{t_i}}(S_{t_i})$ is a random variable. Utilizing the independent increment property of gamma process, we can compute its expectation as follows:

$$\begin{aligned} \mathbb{E}\left[D_{A_{t_i}}(S_{t_i}) \right] &= \int_{Z_{t_i}^+}^{Z_{t_i}^+ + \Delta t} \mathbb{E}\left[\mathbb{1}_{W_{j \geq L} \mid W_{t_i} = w} \right] dj = \int_{Z_{t_i}^+}^{Z_{t_i}^+ + \Delta t} \Pr(W_j \geq L \mid W_{t_i} = w) \, dj \\ &= \int_{Z_{t_i}^+}^{Z_{t_i}^+ + \Delta t} \left(1 - \Pr(W_j - w \leq L - w) \right) dj = \int_{Z_{t_i}^+}^{Z_{t_i}^+ + \Delta t} \left(1 - F_{\Lambda(j) - \Lambda\left(Z_{t_i}^+\right), \beta}(L - w) \right) dj. \end{aligned} \tag{10}$$

In Eq. (10), $F_{\Lambda(j) - \Lambda(Z_{t_i}^+), \beta}(L - w)$ is the cumulative distribution function of the gamma distributed random variable with shape parameter $\Lambda(j) - \Lambda\left(Z_{t_i}^+\right)$ and the scale parameter β. It is evident that the three-argument reward function significantly encapsulates a broader range of information pertaining to the interaction between the agent and the environment, thereby facilitating a more accurate derivation of the optimal value function.

2.5. Optimization metric and algorithm

In this MDP, we try to optimize the average reward instead of the popular total discounted reward metric. This is because in continual tasks where there are no terminal states like the CBM problem, the discounted rewards appear to be problematic (Tadepalli and Ok, 1998). Under a policy π, the average reward ρ_π and the value function V_π of a discrete-time MDP are defined, respectively, as:

$$\rho_\pi = \lim_{N \to \infty} \frac{1}{N} \mathbb{E}_\pi \left[\sum_{i=1}^{N-1} R_i \mid S_0 = \boldsymbol{s} \right], \tag{11}$$

$$V_\pi(\boldsymbol{s}) = \lim_{N \to \infty} \mathbb{E}_\pi \left[\sum_{i=1}^{N-1} (R_i - \rho_\pi) \mid S_0 = \boldsymbol{s} \right]. \tag{12}$$

Equation (12) represents the expected total extra rewards that the agent can receive starting from state $S_0 = \boldsymbol{s}$. The value function for policy π satisfies the following Bellman equation:

$$V_\pi(\boldsymbol{s}) = \int_{\mathcal{S}} \left(r(\boldsymbol{s}, a, \boldsymbol{s}') + V_\pi(\boldsymbol{s}')\right) p(\boldsymbol{s}' \mid \boldsymbol{s}, a) d\boldsymbol{s}' - \rho_\pi. \tag{13}$$

In our specific problem, the reward corresponds to the maintenance cost. Since we define the cost to be non-negative, our objective is to minimize the average cost in the optimization problem. Consequently, the optimal policy $\pi^\star$ is the policy that yields the lowest average cost among all other policies $\rho^\star \leq \rho_\pi, \forall \pi$. And under $\pi^\star$, we obtain the Bellman optimality equation:

$$V^\star(\boldsymbol{s}) = \min_a \left\{ \int_{\mathcal{S}} \left(r(\boldsymbol{s}, a, \boldsymbol{s}') + V^\star(\boldsymbol{s}')\right) p(\boldsymbol{s}' \mid \boldsymbol{s}, a) d\boldsymbol{s}' \right\} - \rho^\star. \tag{14}$$

Once the optimal average cost per action $\rho^\star$ is obtained, we can easily convert it to the average reward per time unit by dividing it by the inter-inspection interval Δt. Note that this only works for periodic inspection case. In this paper, we utilize relative value iteration for continuous state space proposed by Sharma et al. (2020), with some modifications to fit with the problem of interest. The pseudocode for the algorithm is given in Algorithm 1.

Algorithm 1. Model-based approximate relative value iteration for continuous state MDP

Approximate relative value iteration for continuous state MDP

- Input: N evaluating points $S_1, S_2, \dots, S_N \in \mathcal{S}$, a small positive number ϵ; a regression method R: R has two sub methods, a fitting method R_F to fit the data points $\left(S_i, \widehat{V(S_i)}\right), i = 1, \dots, N$, and a prediction method R_P to predict the value of S_i: $R_P(S_i) = \widehat{V(S_i)}$; a positive number K denotes the maximum iteration for the algorithm.
- Initialization: set all $\widehat{V^{(0)}(S_i)}$ to arbitrary values, e.g., zeros, fit $R_F^{(0)}(S_i, \widehat{V^{(0)}(S_i)})$, set $k = 1$ and D to a large positive number.
- While $(D \geq \epsilon)$ and $(k \leq K)$:
  - $\widehat{V_{old}(S_i)} = R_P^{(k-1)}(S_i)$
  - For $i \in (1, \dots, N)$:
    - $\widehat{V_{temp}^{(k)}}(S_i) \leftarrow \hat{T}\widehat{V^{(k-1)}}(S_i)$
    - $\widehat{V^{(k)}}(S_i) \leftarrow \widehat{V_{temp}^{(k)}}(S_i) - \min_j[\widehat{V^{(k)}(S_j)}]$
  - Call the fit method R_F to fit the new data points' values $R_F^{(k)}(S_i, \widehat{V^{(k)}(S_i)})$
  - $D = \max |\widehat{V_{old}(S_i)} - \widehat{R_P^{(k)}(S_i)}|$
  - $k = k + 1$

In this algorithm, $\hat{T}$ is the approximate Bellman operator as the value function is not expressed in an exact form but rather in an approximate form. It is defined as: $\hat{T}V(\boldsymbol{s}) = \min_a \left\{ \int_{\mathcal{S}} \left(r(\boldsymbol{s}, a, \boldsymbol{s}') + \hat{V}(\boldsymbol{s}')\right) p(\boldsymbol{s}' \mid \boldsymbol{s}, a) d\boldsymbol{s}' \right\}$. The superscript (k) in $V^{(k)}$, $R_F^{(k)}$, $R_P^{(k)}$ denote the k-th iteration of the algorithm.

3. Numerical results

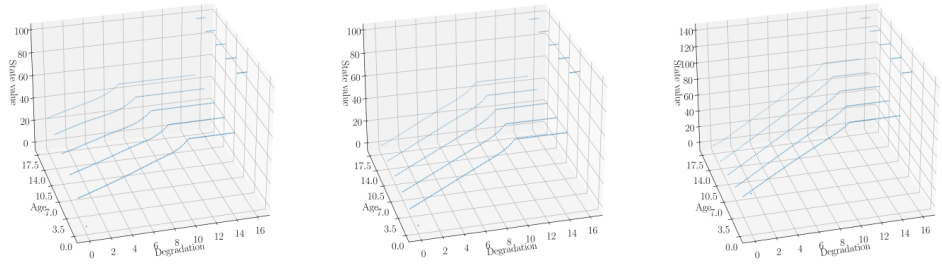
We employ the k-nearest neighbours (KNN) regression (Kramer, 2013) as our chosen regression method. In this approach, we consider a total of $n = 25$ nearest neighbors, and the weight assigned to each neighbor is determined by the distance between the evaluating point and that particular neighbor. To illustrate the dependence of optimal policies on the shape of the degradation process, we utilize three gamma processes with the same rate parameter $\beta = 1$, but different shape functions. Specifically, we use $\Lambda(t) = 0.25t^{1.5}$ for the convex degradation path, $\Lambda(t) = t$ for the linear degradation path, and $\Lambda(t) = 2.5t^{\frac{2}{3}}$ for the concave degradation path. The replacement threshold is $L = 15$. The costs for each action and the downtime cost rate are given in Table 1.

Table 1. Costs of maintenance actions.

Inspection cost c_I	Preventive maintenance cost c_P	Corrective maintenance cost c_C	Downtime cost rate c_D
2	50	100	75

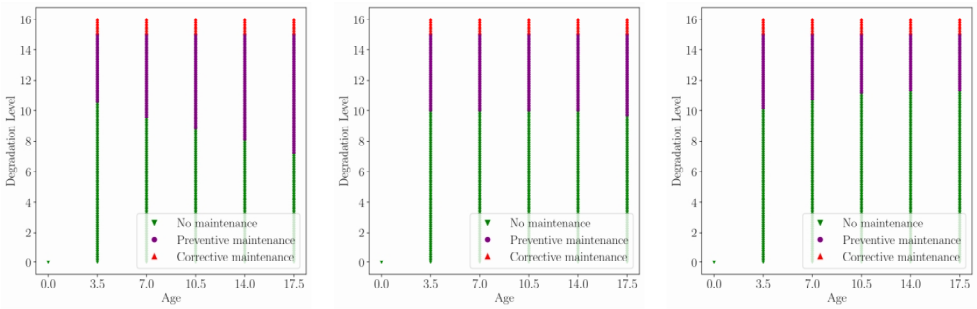
Some examples of the results for $\Delta t = 3.5$ are shown in Figure 4 and 5. In Figure 4, we present the optimal value functions. Since we perform inspections at discrete time intervals of Δt, the age dimension is limited to discrete values such as Δt, $2\Delta t$, and so on. On the other hand, the degradation level for each age can take continuous values. Consequently, the value function can assume continuous values along this dimension. Once we have obtained the optimal value function, we can utilize the Bellman optimality equation (14) to determine

the optimal policy. Figure 5 depicts the optimal action for each state $[W_{t_i}, Z_{t_i}]$, with each marker indicating the recommended action for that particular state. When considering a particular age, it becomes evident that the optimal policy involves implementing preventive maintenance (PM) when the degradation level surpasses a certain optimal threshold (called the PM threshold). Conversely, if the degradation level falls below the threshold, the optimal approach suggests refraining from any maintenance activities (NM). In the linear degradation case, the optimal policy is determined by a single optimal PM threshold for all ages. Therefore, in this scenario, the optimal policy solely relies on the degradation level W_{t_i}. In the convex degradation case, the degradation rate increases significantly with age, resulting in the optimal PM thresholds decreasing as the ages increase. In contrast, in the concave degradation case, the degradation level increases rapidly from an initial, as-good-as-new state. However, over time, the degradation rate slows down. Consequently, the optimal PM threshold for this case decreases as the ages progress.



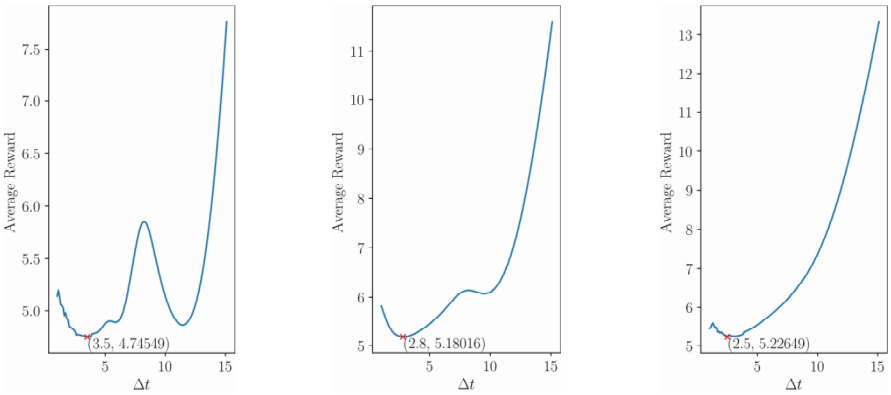
(a) Convex case (b) Linear case (c) Concave case

Fig. 4. Optimal value functions for different degradation shapes with fixed $\Delta t = 3.5$.



(a) Convex case (b) Linear case (c) Concave case

Fig. 5. Optimal policies for different degradation shapes with fixed $\Delta t = 3.5$.



(a) Convex case (b) Linear case (c) Concave case

Fig. 6. Optimal inter-inspection interval for different degradation shape.

For the optimal inter-inspection interval Δt, we conducted searches over different values of Δt on the interval $[1; 15]$. The optimal Δt for each is show in Figure 6. In the case of convex and linear scenarios, there exist two optima for the inspection interval. However, the global optima consistently correspond to the ones with smaller values, thus indicating that condition-based maintenance outperforms time-based maintenance. The alternative optima in these cases consistently align with the optimal policy of replacing the system at the inspection epoch (i.e., time-based maintenance). When considering an increase in the inspection cost, it is important to highlight that the global optimum will progressively rise as well. For instance, in the convex case, let's take the inspection cost $c_I = 5$ as an example. In this case, the optimal inspection interval converges to approximately 12. This observation suggests that when inspection costs are relatively high, favoring a time-based maintenance strategy becomes more advantageous for such scenarios. We conducted a comparison with the widely utilized discretization approach. To simplify the analysis, we focused on a linear degradation path that is solely influenced by the inter-inspection interval Δt, rather than the age. The continuous degradation was discretized into 6 equidistant states. In this discretization scheme, degradation within the interval $[0.0, 2.5)$ corresponds to state 0, $[2.5, 5)$ corresponds to state 1, and so on. The interval $[15.0, \infty)$ is treated as state 6. To solve the discretized states MDP, we utilized the relative Q-learning algorithm for finite state space, as proposed by Gosavi (2013). Furthermore, to establish a baseline for comparison, we employed the well-known parametric approach $(M, \Delta t)$ (e.g., Huynh et al., 2011). The results are presented in Table 2. Within each cell, the number inside the brackets represents the replacement threshold, while the other number denotes the average maintenance cost per time unit. It is worth noting again that for the discretized versions, each state represents an interval, as aforementioned. For instance, state 3 corresponds to the interval $[7.5, 10)$, and if 3 is set as the replacement threshold, preventive maintenance is implemented in every state equal to or greater than 3 (except for the failure threshold 6, which we always conduct preventive maintenance). Additionally, for the continuous state MDP, we provide both the average maintenance cost obtained through the algorithm and the average maintenance cost computed via Monte Carlo simulation. In the majority of cases, the algorithm demonstrates its effectiveness in generating accurate outcomes, as indicated by the relatively small discrepancy between theoretical results and simulations. Moreover, when comparing the discretized state space with the continuous state space, it becomes apparent that preserving the original state space generally results in superior average maintenance costs.

Table 2. Average reward for different inter-inspection interval for continuous and discretized state spaces.

Δt	1	2	3	4	5
Parametric (simulation)	(11.8) 5.81320	(11.2) 5.25047	(10.1) 5.12747	(9.1) 5.22509	(8.0) 5.41636
MDP (continuous, algorithm)	(11.9) 5.99022	(10.8) 5.27072	(9.9) 5.18652	(8.9) 5.27464	(8.0) 5.44408
MDP (continuous, simulation)	(11.9) 5.86753	(10.8) 5.26158	(9.9) 5.18643	(8.9) 5.25839	(8.0) 5.41708
MDP (discretized, simulation)	(5) 6.02045	(4) 5.35443	(4) 5.15926	(4) 5.39509	(3) 5.47418

4. Conclusions and future works

In this paper, we propose an approach to solving the CBM problem modelled as an MDP without discretizing the state-action spaces, thereby preserving their Markov property. Additionally, we have designed the reward function to effectively capture real-world situations. This approach demonstrates significant potential, as it allows for adapting degradation models to meet specific requirements while keeping the solution approach unchanged. The results obtained from the algorithms and the Monte Carlo simulation are in strong agreement.

For future work, given that the maintenance duration in this study is assumed to be negligible, we can further consider this duration to be sufficiently large for practical considerations. Furthermore, it is important to note that the preventive maintenance action is assumed to have a perfect effect, which is not typically the case in most real-world situations. Hence, in future studies, we can consider incorporating an imperfect repair model. Additionally, by changing the inspection period to non-periodic intervals, we can provide the agent with more flexible choices. This modification transforms the problem into a semi-Markov decision process problem.

Acknowledgements

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

References

Adsule, A., Kulkarni, M., Tewari, A. 2020. Reinforcement learning for optimal policy learning in condition-based maintenance. IET Collaborative Intelligent Manufacturing 2(4), 182–188.

Alaswad, S., Xiang, Y. 2017. A review on condition-based maintenance optimization models for stochastically deteriorating system. Reliability Engineering & System Safety 157, 54–63.

Chen, D., Trivedi, K. S. 2005. Optimization for condition-based maintenance with semi-Markov decision process. Reliability Engineering & System Safety 90(1), 25–29.

Chen, N., Ye, Z.-S., Xiang, Y., Zhang, L. 2015. Condition-based maintenance using the inverse Gaussian degradation model. European Journal of Operational Research 243(1), 190–199.

Das, T. K., Gosavi, A., Mahadevan, S., Marchalleck, N. 1999. Solving Semi-Markov Decision Problems Using Average Reward Reinforcement Learning. Management Science 45(4), 560–574.

Gosavi, A. 2013. Relative value iteration for average reward semi-Markov control via simulation. 2013 Winter Simulations Conference (WSC), 623–630.

Huynh, K. T., Barros, A., Berenguer, C., Castro, I. T. 2011. A periodic inspection and replacement policy for systems subject to competing failure modes due to degradation and traumatic events. Reliability Engineering & System Safety 96(4), 497-508.

Kahle, W., Mercier, S., Paroissin, C. 2016. Degradation Processes in Reliability (1st ed.). Wiley.

Kramer, O. 2013. Dimensionality Reduction with Unsupervised Nearest Neighbors. Springer, Berlin, Heidelberg.

Ling, Z., Wang, X., Qu, F. 2018. Reinforcement Learning-Based Maintenance Scheduling for Resource Constrained Flow Line System. 2018 IEEE 4th International Conference on Control Science and Systems Engineering (ICCSSE), 364–369.

Mahadevan, S. 1996. Average Reward Reinforcement Learning: Foundations, Algorithms, and Empirical Results. Machine Learning, 22(1/2/3), 159–195.

Mosayebi Omshi, E., Grall, A. 2021. Replacement and imperfect repair of deteriorating system: Study of a CBM policy and impact of repair efficiency. Reliability Engineering & System Safety, 215, 107905.

Peng, S., Feng, Q. (May). 2021. Reinforcement learning with Gaussian processes for condition-based maintenance. Computers & Industrial Engineering, 158, 107321.

Rasmussen, C. E., Williams, C. K. I. 2008. Gaussian processes for machine learning (3. print). MIT Press.

Schwartz, A. 1993. A reinforcement learning method for maximizing undiscounted rewards. Proceedings of the Tenth International Conference on International Conference on Machine Learning, 298–305.

Sharma, H., Jafarnia-Jahromi, M., Jain, R. 2020. Approximate Relative Value Learning for Average-reward Continuous State MDPs. Proceedings of The 35th Uncertainty in Artificial Intelligence Conference, 956–964.

Sutton, R. S., Barto, A. 2020. Reinforcement learning: An introduction (Second edition). The MIT Press.

Tadepalli, P., Ok, D. 1998. Model-based average reward reinforcement learning. Artificial Intelligence 100(1–2), 177–224.

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

Wang, L., Chu, J., Mao, W. 2008. A condition-based order-replacement policy for a single-unit system. Applied mathematical modelling, 32(11), 2274-2289.

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

Wiering, M., Van Otterlo, M. 2012. Reinforcement Learning in Continuous State and Action Spaces. Van Otterlo, M. (Ed.), Reinforcement Learning: State-of-the-Art 12. Springer Berlin Heidelberg, 211-212.

Xanthopoulos, A. S., Kiatipis, A., Koulouriotis, D. E., Stieger, S. 2018. Reinforcement Learning-Based and Parametric Production-Maintenance Control Policies for a Deteriorating Manufacturing System. IEEE Access 6, 576–588.

Yousefi, N., Tsianikas, S., Coit, D. W. 2022. Dynamic maintenance model for a repairable multi-component system using deep reinforcement learning. Quality Engineering 34(1), 16–35.

Zhang, L., Lei, Y., Shen, H. 2016. How heterogeneity influences condition-based maintenance for gamma degradation process. International Journal of Production Research 54(19), 5829–5841.