Leaf It to Renewal: Improved Predictive Maintenance Policies via Renewal Theory and Decision Trees

Daniel Koutas[1] and Daniel Straub[1]

[1] *Engineering Risk Analysis Group, Technical University of Munich, Bavaria, 80333, Germany*
daniel.koutas@tum.de
straub@tum.de

ABSTRACT

We propose a hybrid planning method for deriving prognostics-based predictive maintenance policies. The method accounts for the available decision options, the information on the future state of the system provided by a prognostic model, and the costs of the underlying renewal-reward process. It results in policies defined by only a few parameters, which can be determined based on theoretical considerations or by optimization from run-to-failure data. We demonstrate the potential of the method in two separate predictive maintenance decision settings: preventive replacement and preventive ordering. Our numerical investigations on a virtual remaining useful life simulator and the C-MAPSS dataset show that the derived policies rival the performance of optimized benchmark policies, while being significantly more robust against overfitting.

1. INTRODUCTION

1.1. Motivation and background

Prognostics is concerned with predicting the future health of a system (Pecht, 2009). Most often, one is interested in estimating the remaining useful life (RUL), i.e., the time left until a specific component or system fails and ceases to perform its intended function (Kim, An, & Choi, 2017). To quantify the uncertainty about the true underlying system state (not available due to, e.g., noisy sensors or hidden faults) as well as the uncertainties associated with making predictions (e.g., aleatory future loads), one typically resorts to a probability distribution over the RUL (Kim et al., 2017; Si, Wang, Hu, & Zhou, 2011). Prognostics is complemented by health management, which concerns making optimal maintenance decisions based on information obtained from the prognostic model (Kim et al., 2017; Nguyen & Medjaher, 2019). This prognostic-based maintenance is summarized with the umbrella term *Prognostics and Health Management* (PHM), or alternatively *predictive maintenance* (PdM) (Ran, Zhou, Lin, Wen, & Deng, 2019).

This contribution focuses on health management, assuming a prognostic model is already available. There are two main options for obtaining maintenance decisions: black box models and heuristics. Black box decision models, such as Deep Reinforcement Learning (DRL) approaches (Koutas, Bismut, & Straub, 2024; Lee & Mitici, 2023; Abbas, Chasparis, & Kelleher, 2022; Jimenez-Roa et al., 2024) often achieve state-of-the-art performance. However, training these models either requires a large corpus of training data or a model that accurately represents the underlying deterioration process. In PHM applications, usually, only limited amounts of training data are available, and there is still a substantial gap between the developed models and the underlying reality (Chao, Kulkarni, Goebel, & Fink, 2022). Another main drawback of DRL approaches lies in the limited transparency and interpretability of the resulting decision policies. In real-life applications, especially for safety-critical systems, this lack of insight is often critical. Therefore, the preferred approach in practice is the use of heuristic policies, i.e., simple rules on how to use the information at hand to choose maintenance actions. Heuristics are guided by intuition and, thus, are easily understood by engineers and operators. Examples for such heuristic policies include probability thresholds and fixed intervals (Kamariotis, Tatsis, Chatzi, Goebel, & Straub, 2024; Luque & Straub, 2019; Hodkiewicz, Bonser, Chin, & While, 2018; He, Tian, Wang, Zuo, & Guo, 2023).

We propose a method to derive heuristic policies by relying on insights from renewal theory (Doob, 1948; Smith, 1958), which generalizes the Poisson process to (almost) arbitrary inter-arrival distributions. By regarding the inter-arrival times as lifetimes of individual components, and the associated failure costs as the rewards, the optimal maintenance problem can be seen as an application of *renewal-reward processes*. Herein, the overarching goal is to find a policy that minimizes the long-running maintenance cost per unit time c_∞.

Daniel Koutas et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.
https://doi.org/10.36001/IJPHM.2026.v17i2.5033

1.2. Related work

Many prognostics-based heuristic maintenance policies have been developed without the use of renewal theory (e.g., Lei & Sandborn, 2016; Asgarpour & Sørensen, 2018; Nguyen & Medjaher, 2019; L. Wang, Zhu, & Zhao, 2024; L. Wang, Chen, Zhao, & Xiang, 2024; Kamariotis et al., 2024). In these works, the parameters of the proposed heuristics are either already fully defined a-priori, or optimized based on available training data. Most commonly, a threshold approach is chosen. This threshold can be defined on various quantities, e.g., on the predicted RUL of a deterministic prognostic model, or on the predicted failure probability within a certain time period.

Previous works (e.g., Zhuang, Xu, & Wang, 2023; Nguyen & Medjaher, 2019; Dong et al., 2025; Y. Wang, Wang, Wu, Zhang, & Wu, 2025; Chen, Shi, Lu, Zhu, & Jiang, 2022) formulate the maintenance optimization problem as the minimization of the cost rate of the monitored components. At first glance, this seems similar to the c_∞ measure of renewal-reward processes. But this cost rate only takes into account the expected costs and lifetime of the monitored component and *not* of the system lifetime and overall component population. In addition, some of the works compute the target as an expectation over individual cost rates, which is *not* equal to the ratio of expected cost over expected lifetime. There are also studies that use c_∞ as a target for making maintenance decisions, such as (Fauriat & Zio, 2020; Mitici, de Pater, Barros, & Zeng, 2023; Yan et al., 2022; Prakash, Narasimhan, & Pandey, 2017). However, these works formulate the long-running maintenance cost per unit time based on the predicted RUL distribution of the single monitored component. By contrast, renewal theory dictates the use of the time-to-failure (TTF) distribution of the whole population, which is evidently different. The use of the component-specific instead of the population-based TTF distribution has been discussed by Kamariotis et al. (2024), who show the negative implications of this choice on the effectiveness of the resulting maintenance policies. Examples for the application of renewal theory to heuristic maintenance strategies on the basis of the whole population of components can be found in the context of optimizing age-based replacement without monitoring, e.g., finding an a-priori optimal age-based replacement time of components (Jiang, 2009). This approach can be expanded in various ways, e.g., to settings with unknown starting component age (Sidibe, Khatab, Diallo, & Kassambara, 2017) or with the inclusion of shocks (Cha, Finkelstein, & Levitin, 2017). Other age-based decision settings are also considered, e.g., for imperfect repairs (Finkelstein, 2015), combinations of repair and replacement (warranties) (Jack, Iskandar, & Murthy, 2009), or ordering (Chelbi & Aït-Kadi, 2001).

Renewal theory has also been used in continuous monitoring settings in which the degradation state of the system can be perfectly observed. In this setting, the asymptotic failure rate of the system subject to a particular stochastic degradation model (e.g., a Gamma process) can be derived, which is then used to find the optimal policy. Examples include the investigation of delayed maintenance operations (Grall, Dieulle, Bérenguer, & Roussignol, 2006), combinations of imperfect as well as perfect repairs (Chien, Sheu, & Zhang, 2012; Mercier & Castro, 2013), or the inclusion of multiple deterioration modes (Saassouh, Dieulle, & Grall, 2007). Again, these studies do not rely on information obtained from a prognostic model.

Similar to continuous monitoring, renewal theory can also be used to derive maintenance decisions based on degradation information obtained through inspections. The majority of these studies consider periodic inspections (De Jonge & Scarf, 2020). Examples include the consideration of imperfect repairs with unavailability constraints (Guo, Wang, Guo, & Si, 2013), the optimization of inspection intervals for a given degradation threshold (Wu, Niknam, & Kobza, 2015), or the inclusion of imperfect inspections (W. Wang & Wang, 2015). The difference between these works and our contribution lies in their limitation to a specific stochastic degradation behaviour (e.g., Wiener or Gamma processes). Moreover, these approaches find an optimal a-priori deterioration threshold or inspection interval, instead of using the most recent RUL distribution for dynamic online decision optimization.

For preventive ordering (our second investigated decision setting), there are some works that perform prognostics-based cost rate minimization to find the optimal ordering time for spare parts, such as in (Elwany & Gebraeel, 2008), or with an extension to stochastic lead times (Z. Wang, Hu, Wang, Kong, & Zhang, 2015). Similar to the discussed works on maintenance decisions, the optimization target of these works is the cost rate of the single monitored component and not the overall ordering cost rate when this policy is applied to the whole population of components.

In addition, there are some works that consider renewal theory-based maintenance for multi-component system. Again, these works do not take into account the information available from a prognostics model. For a more detailed overview, the reader is referred to (De Jonge & Scarf, 2020).

1.3. Contributions and overview

The objective of this paper is to exploit renewal theory to derive prognostics-based heuristic maintenance policies. Our specific contributions are summarized as follows.

- We provide a general method for deriving predictive maintenance policies that incorporate the costs of the underlying renewal process. It consists of constructing a one-step decision tree at each decision time step, where the branch probabilities are computed from the latest RUL distribution, and the branch costs are obtained as the sum of the direct component costs and the costs associated with

continuing the underlying renewal-reward process.

- We implement the proposed method in two decision settings, namely preventive replacement and preventive ordering, for which we provide step-by-step derivations of the resulting maintenance policies.
- We compare the derived maintenance policies against a set of benchmark policies using a virtual RUL simulator and the C-MAPSS dataset. Our derived policies rival the performance of optimized policies, without relying on optimization, and are hence significantly more robust against overfitting in low data regimes.
- We release the RUL simulator as a common reference problem for evaluating and comparing maintenance policies in future studies. The corresponding source code for this work can be found at `github.com/Dakout/leaf_it_to_renewal`.

Besides its improved performance in the low-data regime, the proposed method offers additional advantages over the state of the art. The method merely requires the availability of a prognostic model that outputs RUL distributions. Hence, this work is not constrained to a specific stochastic degradation process. Furthermore, it is applicable across a range of decision settings; additional settings beyond those investigated in the paper are listed in the discussion section.

The remainder of this paper is structured as follows. Section 2 introduces the general framework of predictive maintenance and the two decision settings investigated in this work. Section 3 proposes a new method for deriving predictive maintenance heuristics and applies it to the introduced decision settings, together with a set of benchmark heuristic policies. Section 4 conducts a detailed investigation of the proposed policies and benchmarks using a virtual RUL simulator. Section 5 validates the obtained results on the prominently used C-MAPSS dataset. Section 6 provides a thorough discussion on the limitations of the derived heuristics, as well as suggestions for further applications of the proposed method. Section 7 concludes this paper with a short summary.

2. DECISION SETTING AND NOTATION

We adopt large parts of the setting from (Kamariotis et al., 2024), i.e., we consider a single component subject to continuous monitoring, which is used to periodically make a probabilistic RUL prediction. This prediction is then fed into a maintenance policy, which outputs a decision to either do nothing or to perform a maintenance action[1]. The workflow is illustrated in Figure 1.

In this paper, we consider and discuss two PdM decision settings:

1. Pure replacement: one decides when to replace a component subject to two possible types of replacement, namely preventive replacement with cost c_p and corrective replacement upon failure with cost c_c, where $c_c > c_p$ to represent higher damages or longer downtime. It is assumed that a spare component is available.
2. Pure ordering: one decides when to order a new component, which arrives after a deterministic lead time l, considering unavailability c_{unav} and inventory holding c_{inv} cost rates, where $c_{\text{unav}} > c_{\text{inv}}$. The component is replaced following failure as soon as the spare is available. In this paper, for simplicity, it is assumed that replacement of the failed component incurs no additional costs, but this assumption can easily be removed.

t is the time since a component has been installed. The set of discrete decision times for ordering/replacement is denoted with $\{t_k = k \cdot \Delta t, k = 1, 2, \dots\}$. Further assumptions are that failures are self-announcing, replacements are perfect, instantaneous (take 0 time) and are performed immediately upon failure of the component. Furthermore, the lifetimes of all components are independent and identically distributed (with finite mean). Hence, we can use renewal-reward theory, where a renewal cycle is defined by the time interval between two successive replacements. Finally, we do not consider discounting of future costs, assuming that component lifetimes are comparably short.

Preventive replacement, order and failure times of a component are denoted with T_{rep}, T_{ord}, and T_F, respectively. If reference to a specific sample is needed, we explicitly refer to it via a sample superscript for component i, e.g., $T_F^{(i)}$; otherwise, the sample-based notation is avoided wherever possible for the sake of readability. We further introduce relative time $r_k = t - t_k$, which follows the definition of the remaining useful life $RUL(t) = T_F - t$, where $RUL_k \coloneqq RUL(t_k)$. For the derivations and definitions of the proposed heuristic maintenance policies in this work, both notations (r vs. t & RUL vs. T_F) will be used to reduce verbosity of the formulas as well as to facilitate intuition.

To evaluate the average performance of a certain maintenance policy, it needs to be repeatedly applied to numerous components that follow some failure time distribution. Upon failure/replacement, the component is assumed to be as good as new, therefore, we can use the elementary renewal-reward theory in both decision settings to compute the policy's long-running maintenance cost per unit time, c_∞, as (Grimmett & Stirzaker, 2020)

$$c_\infty = \lim_{t\to\infty} \frac{C(t)}{t} = \frac{\mathbb{E}[C]}{\mathbb{E}[T_{\text{lc}}]}. \tag{1}$$

The expectations in the numerator and denominator of Eq. (1) are both taken based on the resulting renewal cycles of the

[1] Here, a maintenance action refers to any action that helps to keep the system in a working condition, and, therefore, includes replacement as well as ordering.

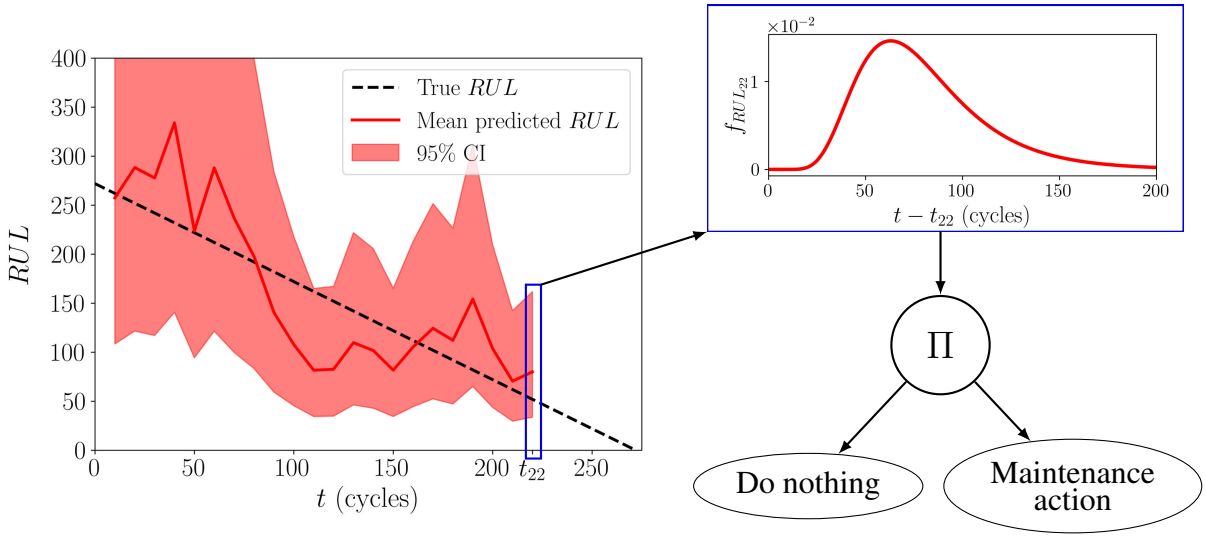


Figure 1. General workflow of maintenance decision policies. At every time step, the latest RUL-PDF is fed into a heuristic policy (Π), which outputs a decision to either do nothing or to perform a maintenance action, e.g., replacement or ordering a spare component. The workflow is exemplarily shown for the time step $t_k = 220$.

component, and C and T_{lc} are determined by the specific policy of the respective decision settings. For the replacement setting, C is either c_p or c_c, and T_{lc} denotes either the point in time when the component is preventively replaced or fails, whichever happens first. For the ordering setting, C is either the accumulated inventory or unavailability cost, and T_{lc} is always the component's failure time. Hence, c_∞ refers to $\frac{\mathbb{E}[C_{\text{rep}}]}{\mathbb{E}[T_{\text{lc}}]}$ or $\frac{\mathbb{E}[C_{\text{ord}}]}{\mathbb{E}[T_{\text{lc}}]}$, depending on the decision context. The overarching goal is to find policies which minimize c_∞.

The investigated policies are evaluated by computing a Monte Carlo (MC) estimate of the relative deviation between their cost rate and the optimal policy's cost rate. This metric was introduced as $\hat{M}$ in (Kamariotis et al., 2024):

$$\hat{M} = \frac{\hat{C}_\infty - \hat{C}^*_\infty}{\hat{C}^*_\infty}, \qquad 0 \leq \hat{M}. \tag{2}$$

A higher value of $\hat{M}$ indicates a greater deviation from the optimal policy, and, consequently, poorer performance. $\hat{C}_\infty$ and $\hat{C}^*_\infty$ are MC estimates of the long-running cost rate for an investigated policy and the perfect policy, respectively. They are defined as (Kamariotis et al., 2024)

$$\hat{C}_\infty = \frac{\frac{1}{n}\sum_{i=1}^{n} C^{(i)}}{\frac{1}{n}\sum_{i=1}^{n} T_{\text{lc}}^{(i)}}, \tag{3}$$

$$\hat{C}^*_\infty = \frac{\frac{1}{n}\sum_{i=1}^{n} C^{(i)^*}}{\frac{1}{n}\sum_{i=1}^{n} T_{\text{lc}}^{(i)^*}}. \tag{4}$$

C^* and T^*_{lc} are the maintenance costs and component lifetimes resulting from a policy that has perfect foresight of each component's failure time and therefore can act optimally. For a given decision setting, they can be computed as the arguments that minimize a component's cost rate. For example, in the preventive replacement setting, the two options are 1) let the component fail or 2) replace at the last decision time before failure:

$$\frac{C^{(i)^*}}{T_{\text{lc}}^{(i)^*}} = \min\left(\frac{c_c}{T_F^{(i)}}, \frac{c_p}{\left\lfloor \frac{T_F^{(i)}}{\Delta t} \right\rfloor \cdot \Delta t}\right). \tag{5}$$

Likewise, in the preventive ordering setting, the lifetimes are equal to the failure times: $T_{\text{lc}}^{(i)^*} := T_F^{(i)}$, and the ordering times are chosen such that the inventory/unavailability costs are minimized:

$$C^{(i)^*} = \min\Big(c_{\text{inv}} \cdot \left[T_F^{(i)} - T_{\text{inv}}^{(i)^*}\right], \\ c_{\text{unav}} \cdot \left[T_{\text{inv}}^{(i)^*} + \Delta t + l - T_F^{(i)}\right]\Big), \tag{6}$$

where $T_{\text{inv}}^{(i)^*} = \left\lfloor \frac{T_F^{(i)} - l}{\Delta t} \right\rfloor \cdot \Delta t$, and $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$ represent the floor and ceiling functions, respectively.

In the considered setting, one obtains from a prognostic model at each t_k a predicted RUL probability density function (PDF) denoted with $f_{RUL_k}(r)$. This prediction is based on all monitoring information available up to time t_k. This distribution is the input to the decision policy, as illustrated in Figure 1. Throughout this work, $f_{RUL_k}(r)$ is used as the reference dis-

tribution to compute cumulative probabilities and expectations. In particular, the probability of the RUL being in the interval $(a, b]$, Eq. (7), and the expected RUL given that it lies in the interval $(a, b]$, Eq. (8) are obtained as

$$\Pr(a < RUL_k \leq b) \coloneqq \int_a^b f_{RUL_k}(r)\ \mathrm{d}r \tag{7}$$

$$\mathbb{E}\left[RUL_k \mid a < RUL_k \leq b\right] \coloneqq \frac{1}{\Pr(a < RUL_k \leq b)} \cdot \int_a^b r \cdot f_{RUL_k}(r)\ \mathrm{d}r. \tag{8}$$

The decision interval Δt is highly problem-dependent. As an example, for aero-engines, Kamariotis et al. (2024) suggest that $\Delta t = 5 - 10$ flight cycles is realistic.

The available actions in the two decision settings, "do nothing", "preventively replace", and "preventively order", are abbreviated as DN, PR, and PO, respectively.

3. PDM POLICIES

In a general maintenance setting, individual decisions affect the state of the underlying system; they also impact future decision alternatives. Hence, one must consider the full sequence of decisions over the entire structure's life to quantify the effect of individual decisions. Because of uncertainty in the past, present, and future states and available information, this setting corresponds to a stochastic sequential decision problem (SDP) (Raiffa & Schlaifer, 2000; Kochenderfer, 2015; Bismut & Straub, 2021). The full decision tree grows polynomially/(super-)exponentially in the set of the number of system states, observations, actions, components and considered time horizon, which is referred to as *curse of dimensionality* (Bellman, 1966) and *curse of history* (Pineau, Gordon, & Thrun, 2006).

The decision on which action to take at time step t_k is determined by a function called *policy* π_{t_k}. For its definition, we adopt the notation from Markov decision processes (MDPs) (Sutton & Barto, 1998). Given a set of component states $s \in \mathcal{S}$ and available actions $a \in \mathcal{A}$, a policy assigns a probability distribution over $\mathcal{A}$, denoted as $\pi_{t_k}(a \mid s) = \Pr(a \mid S_{t_k} = s)$[2]. Thus, policies are generally stochastic. However, in this paper, we consider only deterministic policies, denoted as $\pi_{t_k}(s)$, which prescribe an action a to any state s (Kaelbling, Littman, & Moore, 1996).

In practical cases, the true underlying state of the component is typically unknown, i.e., we have state uncertainty. Following the notation of *partially observable* MDPs (POMDPs), this state uncertainty is represented by *belief states* $b \in \mathcal{B}$, which are probability distributions over $\mathcal{S}$. If $b_{t_k}(s) = \Pr(S_{t_k} = s)$ denotes the probability of the system being in state s at time t, then by the axioms of probability theory we obtain the following constraints: $0 \leq b_{t_k}(s) \leq 1,\ \forall s \in \mathcal{S}$ and $\sum_{s\in\mathcal{S}} b_{t_k}(s) = 1$. In POMDPs, the policy can be reformulated to be a function of the belief state: $\pi_{t_k}(b)$ (Kaelbling, Littman, & Cassandra, 1998). This belief is updated with Bayes' rule, taking into account the underlying system dynamics.

For our predictive maintenance formulation, we define the true remaining useful life as the state of the system, and the RUL-PDF obtained by the prognostic model as the belief state:

$$S_{t_k} \coloneqq RUL_k \tag{9}$$

$$b_{t_k} \coloneqq f_{RUL_k}\ . \tag{10}$$

This RUL-PDF is the input to all considered policies, which is why it is omitted in the following; we simply write $\pi_{t_k} = \pi_{t_k}(b)$. Note that the RUL-PDF is provided by a prognostic model, whose choice and training are deliberately left to the user. Thus, the resulting belief representation does not necessarily coincide with the Bayesian belief over the underlying degradation state, and the final maintenance performance is inherently conditioned on the predictive quality of the employed prognostic model.

A *strategy* $\Pi = \{\pi_{t_1}, \pi_{t_2}, \dots\}$ denotes the set of policies for all time steps, where $\mathcal{P}$ is the space of all strategies. The optimal strategy Π^* in the context of renewal-reward theory is then the one which minimizes the long-running maintenance cost per unit time:

$$\Pi^* = \underset{\Pi\in\mathcal{P}}{\arg\min}\ \left[c_\infty \mid \Pi\right]. \tag{11}$$

Examining all possible Π is computationally unfeasible; hence, one needs a way to reduce the strategy space $\mathcal{P}$. We do this by using a *heuristic*, which is a parametrized strategy $\Pi_{\mathbf{w}}$ with parameters $\mathbf{w} = \{w_1, w_2, \dots, w_h\}$. Instead of Eq. (11), we thus solve the following optimization problem (Bismut & Straub, 2021):

$$\mathbf{w}^* = \underset{\mathbf{w}}{\arg\min}\ \left[c_\infty \mid \Pi_{\mathbf{w}}\right]. \tag{12}$$

Since $\Pi_{\mathbf{w}}$ forms a subspace of $\mathcal{P}$, solving Eq. (12) does not necessarily coincide with the global optimum obtained by solving Eq. (11).

We consider an infinite time horizon, assuming that components are continuously replaced. In reality, the system will have a finite lifetime, but if the component life is signifi-

[2]This notation already assumes that the state s is Markovian (Ethier & Kurtz, 2009); otherwise, the policy depends on the full history of states and actions: $\pi_{t_k}(s_{0:t_k}, a_{0:t_{k-1}})$ (Kochenderfer, 2015).

cantly shorter than the system life, the infinite horizon is a good approximation (Hinderer & Waldmann, 1999). Because of the infinite horizon, the optimal strategies are *stationary*, i.e., the policies at each time step are identical. We thus further simplify the policy notation by dropping the time index $\Pi = \pi = \pi_{t_1} = \pi_{t_2} = \dots$ (Bismut & Straub, 2021). For this reason, in this work, we use the terms policy and strategy interchangeably.

3.1. Proposed method for deriving PdM policies

Heuristic policies can be viewed as a way to simplify the underlying complexity of the full decision tree. In this paper, we propose a specific method to derive predictive maintenance policies by considering only the one-step decision tree. This corresponds to an online policy planning with a receding horizon of depth 1. Such a one-step formulation can still be quite effective, as frequent replanning can compensate for the lack of longer-term modeling (Kochenderfer, Wheeler, & Wray, 2022).

The motivation behind this originates in real-life settings, where maintenance actions (such as order, inspect, repair, replace) cannot be performed at arbitrary times, but are limited to a set of available options. As an example, for the inspection of an aircraft, the restrictions resulting from the planned operation time of the aircraft, as well as the availability of the crew, must be taken into account. In our setting, this restriction is represented by the decision interval Δt. The steps for our proposed method are enumerated as follows.

i. Construct a simplified decision tree based on the available options and potential consequences.
ii. Assign probabilities to each branch based on the RUL-PDF obtained by the prognostic model.
iii. Assign to each branch the expected costs resulting from
   a) the monitored component,
   b) the underlying renewal-reward process.
iv. Simplify the cost expressions and derive the policy that chooses the action with minimal expected cost according to the simplified decision tree.
v. Optional: Perform data-based optimization of selected parameters of the derived policy.

In step iii, renewal theory is used to quantify the expected benefits associated with keeping the component in service longer. Specifically, we use the long-running cost per unit time of the respective maintenance policy (c_∞). If the component is replaced by a new component at time t_k instead of $t_k + \Delta t$, this will incur an expected cost that is equal to c_∞ times Δt. This choice can be justified in two different ways. Firstly, the time-to-failure and maintenance costs of the future components are unknown; c_∞ serves here as an expected value. Secondly, the assumption for the use of renewal theory is that the underlying renewal-reward process continues in the limit until infinity. Herein, c_∞ represents the "true" cost rate of the maintenance policy.

The core of the proposed heuristic policies is the simplification of the full sequential decision problem into the binary one-step decision of *perform an action now or perform an action later*. As a representation for *act later*, we choose the action at the next possible decision time step t_{k+1}. Our reasoning behind this is that if it is optimal to perform the action at some later stage $t_{k+\tau}$, then we assume that it is better to act at t_{k+1} as opposed to acting at t_k. Thus, ultimately, we consider the decision problem of acting immediately at t_k versus at the next possible decision time t_{k+1}. The validity of this assumption is discussed in more detail in Section 6.

If the heuristic decision rule yields that it is better to act at t_k, then the action is performed at t_k. On the other hand, if the heuristic favours an action at t_{k+1}, then this does not necessarily mean that the action is actually performed at t_{k+1}; instead, the decision is revisited at t_{k+1} following the online planning paradigm. At t_{k+1}, one obtains new information from the prognostic model and repeats the evaluation of the two decision options. This simplified binary decision can be represented by a one-step decision tree, which will be used throughout this paper.

We refer to the heuristic policies derived with the proposed method as *discrete option assessment* policies and correspondingly use doa to refer to these policies. In the following Sections 3.2 and 3.3, we derive doa policies together with different benchmarks for the preventive replacement and preventive ordering settings.

3.2. Heuristic replacement policies

In this section, we consider the pure replacement setting. First, two benchmark heuristic policies are introduced in Sections 3.2.1 and 3.2.2, which are taken from (Kamariotis et al., 2024). Afterwards, we present a new replacement heuristic based on the one-step decision tree in Section 3.2.3.

3.2.1. Reference heuristic: simple threshold

The first heuristic replacement policy, termed $rh1$, is a standard strategy used frequently in the literature for risk-based inspection planning (e.g., Luque & Straub, 2019; Nielsen & Sørensen, 2018)) and PHM (e.g., Kamariotis et al., 2024; Nguyen & Medjaher, 2019; He et al., 2023)). $rh1$ performs a replacement action whenever the predicted probability of failure in the next time interval exceeds a threshold value p_{thres} (this threshold is the sole parameter w in this heuristic):

$$a_k = \Pi^{rh1}_{p_{\text{thres}}} = \begin{cases} \text{PR}, & \text{if } p_{\text{thres}} < \Pr\left(RUL_k \leq \Delta t\right) \\ \text{DN} & \text{else.} \end{cases} \tag{13}$$

A default choice for the probability threshold is $p_{\text{thres}} = c_p/c_c$[3]. However, this choice is suboptimal and can perform poorly (Kamariotis et al., 2024). If training data is available, it is preferable to find p^*_{thres} by solving the optimization problem in Eq. (12).

3.2.2. Reference heuristic: optimal replacement time based on full RUL-PDF

As another benchmark, we consider *PdM policy 3* by Kamariotis et al. (2024), which we denote by $rh2$. At each decision step, an optimization problem is solved to determine the optimal replacement time. The corresponding decision rule is

$$a_k = \Pi^{rh2}_{c_\infty} = \begin{cases} \text{PR}, & \text{if } r^*_{\text{rep},k} < \Delta t \\ \text{DN} & \text{else,} \end{cases} \tag{14}$$

where the optimal replacement time $r^*_{\text{rep},k}$ is defined as

$$r^*_{\text{rep},k} = \arg\min_{r_{\text{rep}}} p_{F,k}(r_{\text{rep}}) \cdot c_c + (1 - p_{F,k}(r_{\text{rep}})) \cdot c_p + c_\infty \cdot \int_{r_{\text{rep}}}^{\infty} (r - r_{\text{rep}}) \cdot f_{RUL_k}(r)\,\mathrm{d}r, \tag{15}$$

with $p_{F,k}(r_{\text{rep}}) = \Pr(RUL_k \leq r_{\text{rep}})$. The idea behind $rh2$ is to not only consider the costs resulting from preventive/ corrective replacement of the monitored component, but also to include the costs incurred from replacing the component too early. The cost rate for this "opportunity cost" is set to c_∞, following the logic explained in Section 3.1. However, the true value of c_∞ is unknown, since the final policy is the accumulation of individual maintenance decisions that are based on c_∞ themselves. To escape this implicit definition, we use an estimate for this value, denoted as $c_{\infty,0}$. Three options to obtain such an estimate are presented in (Kamariotis et al., 2024). In this work, we only use the first option, which considers a situation without monitoring (NM), where renewal theory can be used to find the optimal replacement time (component age). The objective function for a deterministic replacement time t_{rep} is formulated as:

$$g^{NM}(t_{\text{rep}}) = \frac{p_F(t_{\text{rep}}) \cdot c_c + (1 - p_F(t_{\text{rep}})) \cdot c_p}{\int_0^{t_{\text{rep}}} t \cdot f_{T_F}(t)\mathrm{d}t + (1 - p_F(t_{\text{rep}})) \cdot t_{\text{rep}}}, \tag{16}$$

with $p_F(t_{\text{rep}}) = \int_0^{t_{\text{rep}}} f_{T_F}(t)\mathrm{d}t$. The corresponding initial estimate for the cost rate is found by solving the optimization problem:

$$c_{\infty,0} = \min_{t_{\text{rep}}} g(t_{\text{rep}}), \quad t_{\text{rep}} \in (0, \infty). \tag{17}$$

We have also performed detailed investigations on the performance of alternative versions of $rh2$, the results of which are given in Appendix D.

3.2.3. Proposed policy

We derive a new doa preventive replacement heuristic with the proposed method of Section 3.1. Since we show all steps for a generic time step t_k, we omit the subscript k in the derivation for ease of notation, and subsequently reintroduce it in the final policy expression. In step i, we construct a one-step decision tree for the time horizon of Δt, depicted in Figure 2. There are two decision alternatives (DN: do nothing; PR: preventive replacement) at the current time. Following PR, a new component will be in place during Δt. This new component is not explicitly modeled. If the do-nothing action is chosen, the component can fail within Δt or be safe. In total this leads to four branches for this one-step decision tree, which are labelled (a) to (d) in Figure 2. The probabilities associated with the two branches following the DN action are readily obtained (step ii); the probability of failure during Δt is $p_F = \Pr(RUL \leq \Delta t)$.

Figure 3 illustrates the relation between the predicted RUL distribution, the failure time, and the resulting cost for branches (a) to (c). Following the proposed method, we do not assign costs to branch (d), since we restrict the maintenance options to t_k and t_{k+1}, and do not consider the replacement options at $t_{k+\tau}, \tau > 1$ when establishing the policy for time t_k. In the following, Figure 3 forms the basis for establishing the expected costs associated with branches (a) to (c).

Following Section 3.1, we use c_∞ for quantifying the costs of the continuation of the renewal-reward process upon failure or replacement of the component. If $c_{M@t_k}$ denotes the expected costs of performing maintenance action M at time t_k, then the expected costs for preventively replacing the component immediately, i.e., branch/subfigure (a), is obtained as

$$c_{PR@t_k}(r) = c_p + c_\infty \cdot r. \tag{18}$$

Here, r denotes the considered time horizon.

The expected cost of planning to replace at the next decision time is a function of expected failure costs, i.e., branch (b),

$$c_{DN@t_k, T_F \leq t_{k+1}}(r) = c_c + c_\infty \cdot (r - \mathbb{E}[RUL \mid RUL < \Delta t]) \tag{19}$$

as well as the expected preventive replacement costs, i.e.,

[3]This threshold can be derived by comparing the preventive maintenance costs c_p with the direct expected failure costs until the next decision time $\Pr(RUL_k \leq \Delta t) \cdot c_c$.

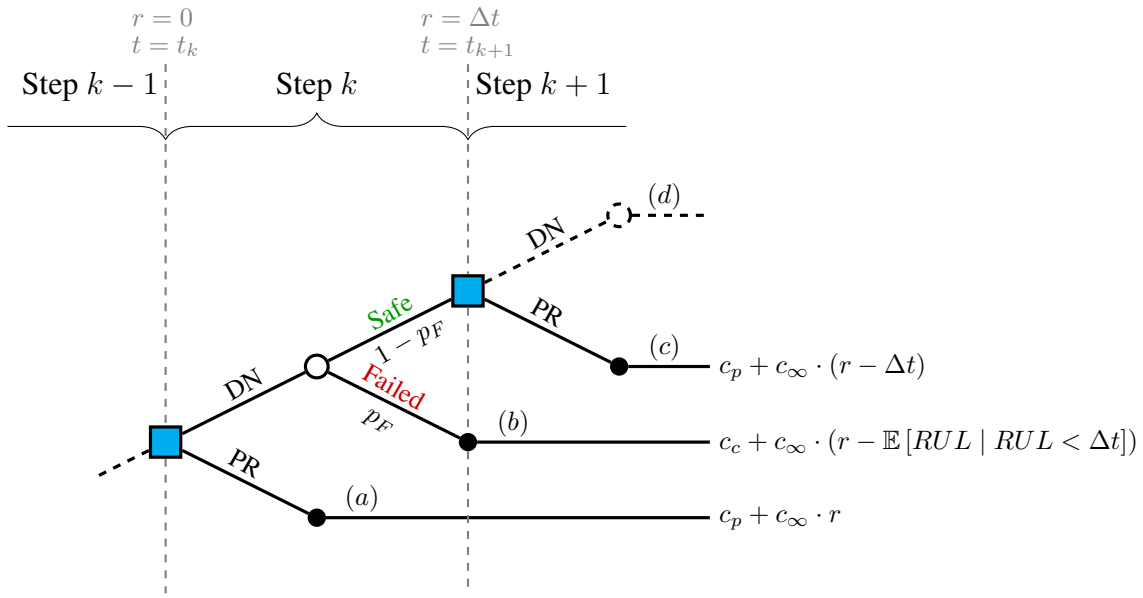


Figure 2. Decision tree for a deteriorating component that can be preventively replaced. Component states are represented by round nodes, and maintenance decisions (a_k) as square nodes. The action taken (PR/DN) as well as the condition of the component (Failed/Safe) with the corresponding event probabilities are included in the respective paths, where $p_F = \Pr(RUL \leq \Delta t)$. The component's end of life (filled black circle) is shown together with the associated expected costs. The Figure is adapted from (Bismut & Straub, 2021).

branch (c),

$$c_{DN@t_k, PR@t_{k+1}}(r) = c_p + c_\infty (r - \Delta t). \tag{20}$$

Since replacement at other points in time is not considered (branch d), the total expected cost of doing nothing at the current time step (planning to replace at t_{k+1}) is

$$\begin{aligned} c_{PR@t_{k+1}}(r) = \Pr(RUL \leq \Delta t) \cdot c_{DN@t_k, T_F \leq t_{k+1}}(r) \\ + \Pr(RUL > \Delta t) \cdot c_{DN@t_k, PR@t_{k+1}}(r). \end{aligned} \tag{21}$$

By comparing the expected costs for the two available decision options Eqs. (18) and (21), we derive the heuristic replacement rule by choosing the action with minimal expected cost. This is expressed as

$$a_k = \Pi^{doa}_{c_\infty, r} = \begin{cases} \text{PR}, & \text{if } c_{PR@t_k}(r) < c_{PR@t_{k+1}}(r) \\ \text{DN} & \text{else.} \end{cases} \tag{22}$$

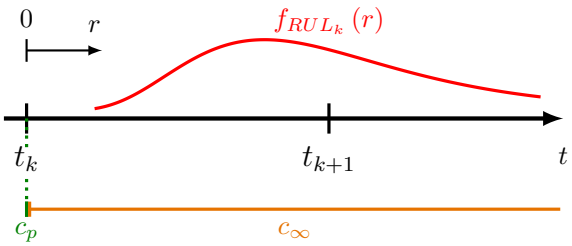
Inserting Eqs. (18) to (21) into the decision heuristic into Eq. (22), results - after some algebra - in the final *doa* replacement decision heuristic

$$a_k = \Pi^{doa}_{c_\infty} = \begin{cases} \text{PR}, & \text{if } c_\infty \Delta t < \Pr(RUL_k \leq \Delta t) \cdot \\ & \quad [c_c - c_p + c_\infty \cdot (\Delta t \\ & \quad - \mathbb{E}[RUL_k \mid RUL_k \leq \Delta t])] \\ \text{DN} & \text{else.} \end{cases} \tag{23}$$

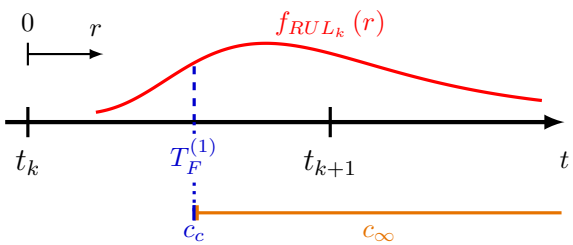
The full derivation of Eq. (23) is given in Appendix A.1. Note that this policy does not depend on r, hence there is no need to choose this value.

Eq. (23) offers a nice, intuitive interpretation: One should preventively replace the component only if the guaranteed costs incurred from the continuation of the renewal-reward process ($c_\infty \cdot \Delta t$) are smaller than the expected costs of failure in the interval $[t_k,\ t_{k+1}]$. These failure costs are composed of the additional replacement costs ($c_c - c_p$) as well as the costs associated with the continuation of the renewal-reward process, reduced by the conditional expected remaining useful life.

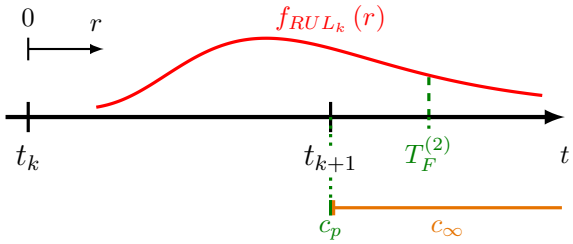
Following $rh2$, we again use the estimate $c_{\infty,0} \approx c_\infty$ presented in Section 3.2.2 for the heuristic policy. Instead of this estimate, one can also regard c_∞ as a mere parameter, and perform data-driven optimization by solving Eq. (12) with respect to the final policy performance $\hat{C}_\infty$, following step v of the general method. We investigate both options numerically in Section 4.



(a) Preventive replacement at t_k



(b) Do nothing at t_k, failure before t_{k+1}



(c) Do nothing at t_k, preventive replacement at t_{k+1}

Figure 3. Exemplary incurred costs for each of the considered branches (a) to (c) of the decision tree in Figure 2.

3.3. Heuristic ordering policies

In this section, we use the proposed method to develop a policy for preventive ordering. We assume here that if a spare component is in stock, the component can be replaced immediately upon failure. For holding a component in stock, one pays the inventory holding cost:

$$C_{\text{stock}} = \max\left(T_F - (T_{\text{ord}} + l),\ 0\right) \cdot c_{\text{inv}}. \tag{24}$$

If a spare component is ordered too late, the component cannot be replaced, and one pays a delay cost of

$$C_{\text{delay}} = \max\left(T_{\text{ord}} + l - T_F,\ 0\right) \cdot c_{\text{unav}}. \tag{25}$$

If a monitored component fails before an order is placed, an ordering is enforced at the next possible order time $T_{\text{ord}} = \left\lceil \frac{T_F}{\Delta t} \right\rceil \cdot \Delta t$. Unavailability costs are accumulated while waiting for the component, and the renewal-reward process continues once the component has arrived after the lead time l.

Next, we introduce a benchmark heuristic ordering rule and then derive the heuristic ordering policy according to our proposed method in Section 3.3.2.

3.3.1. Reference heuristic: simple threshold

For benchmarking, we formulate a simple heuristic policy, which we denote with $oh1$. It requires ordering if the probability of failure after a time period $v = \left\lceil \frac{l}{\Delta t} \right\rceil \cdot \Delta t$ exceeds a threshold:

$$a_k = \Pi^{oh1}_{p_{\text{thres}}} = \begin{cases} \text{PO}, & \text{if } p_{\text{thres}} < \Pr\left(RUL_k \leq v\right) \\ \text{DN} & \text{else,} \end{cases} \tag{26}$$

in which the threshold p_{thres} is the heuristic parameter, which can be optimized or set to the default value $c_{\text{inv}}/c_{\text{unav}}$.

3.3.2. Proposed policy

Following our proposed method, we establish a one-step decision tree in which we consider two options: either order a spare component now at t_k or at the next possible decision time t_{k+1}. The decision tree with associated branch probabilities and expected costs (steps i. to iii.) is shown Figure 4. The branches are again labelled (a) to (d). Example realizations of these four branches are shown in Figure 5, with coinciding subfigure labels. Again, we omit the subscript k in the derivation of the policy for ease of notation, and then reintroduce it in the final policy expression.

The long-running maintenance cost per unit time, c_∞, is again used to account for the costs associated with the continuation of the renewal-reward process after a component has been replaced. Based on this, the expected costs associated with the branches (a) - (d), corresponding to Figures 5a to 5d are given in Eqs. (27) to (30):

$$\begin{aligned} c_{O@t_k, T_F \leq t_k + l}(r) = (r - l) \cdot c_\infty \\ + \left(l - \mathbb{E}\left[RUL \mid RUL \leq l\right]\right) \cdot c_{\text{unav}} \end{aligned} \tag{27}$$

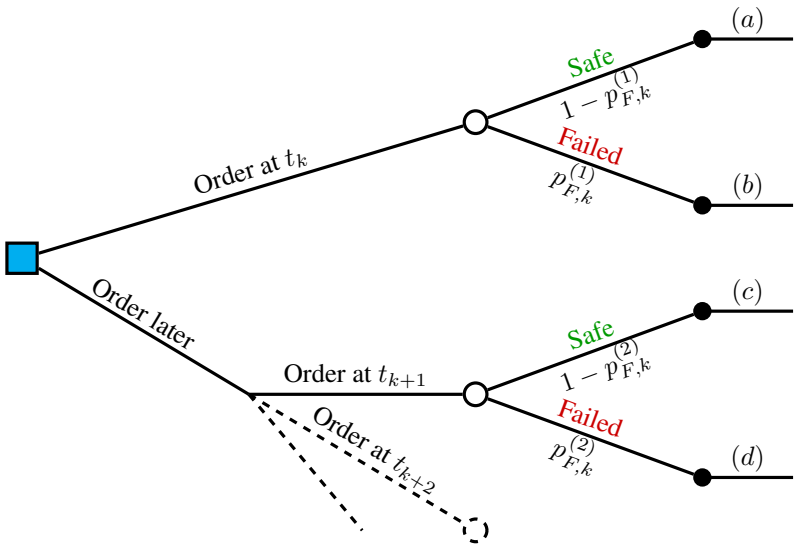


Figure 4. Decision tree for a deteriorating component, where a spare can be preventively ordered. Component states are represented by round nodes and the ordering decision (a_k) as a square node. The action taken, as well as the condition of the component (Failed/Safe) with the corresponding event probabilities are included in the respective paths, where $p_F^{(1)} = \Pr(RUL \leq l)$ and $p_F^{(2)} = \Pr(RUL \leq l + \Delta t)$. Here, filled black circles indicate the component's eventual end of life (the ordering decision has no influence on that). The costs for each branch are not shown in the graph to avoid cluttering; instead, they are listed in Eqs. (27) to (30).

$$c_{O@t_k, T_F > t_k + l}(r) = (r - \mathbb{E}[RUL \mid RUL > l]) \cdot c_\infty + (\mathbb{E}[RUL \mid RUL > l] - l) \cdot c_{\text{inv}} \quad (28)$$

$$c_{O@t_{k+1}, T_F \leq t_{k+1} + l}(r) = (r - l - \Delta t) \cdot c_\infty + (l + \Delta t - \mathbb{E}[RUL \mid RUL \leq l + \Delta t]) \cdot c_{\text{unav}} \quad (29)$$

$$c_{O@t_{k+1}, T_F > t_{k+1} + l}(r) = c_\infty \cdot (r - \mathbb{E}[RUL \mid RUL > l + \Delta t]) + c_{\text{inv}} \cdot (\mathbb{E}[RUL \mid RUL > l + \Delta t] - l - \Delta t). \quad (30)$$

The resulting total expected costs of the decisions "ordering at t_k" and "ordering at t_{k+1}" are

$$c_{O@t_k}(r) = \Pr(RUL \leq l) \cdot c_{O@t_k, T_F \leq t_k + l}(r) + \Pr(RUL > l) \cdot c_{O@t_k, T_F > t_k + l}(r) \quad (31)$$

$$c_{O@t_{k+1}}(r) = \Pr(RUL \leq l + \Delta t) \cdot c_{O@t_{k+1}, T_F \leq t_{k+1} + l}(r) + \Pr(RUL \leq l > \Delta t) \cdot c_{O@t_{k+1}, T_F > t_{k+1} + l}(r). \quad (32)$$

The heuristic ordering policy is to choose the action with minimal expected cost (step iv):

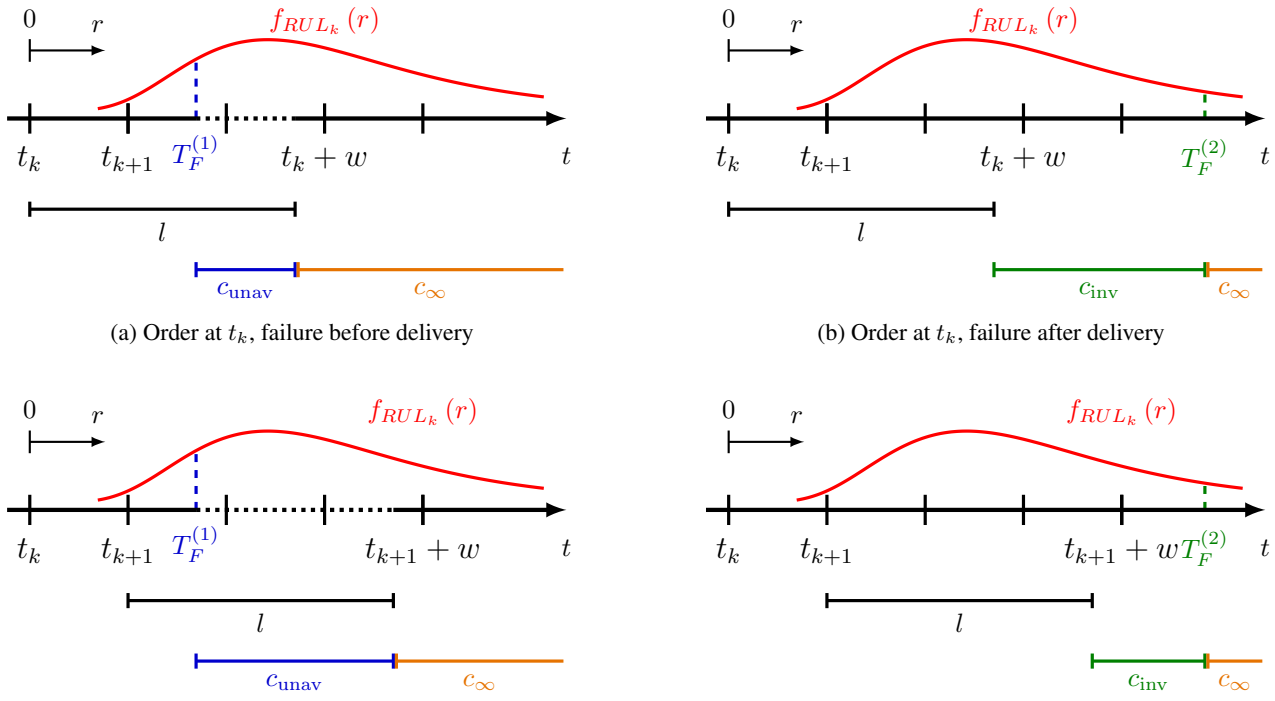
$$a_k = \Pi_{c_\infty, r}^{doa} = \begin{cases} \text{PO}, & \text{if } c_{O@t_k}(r) < c_{O@t_{k+1}}(r) \\ \text{DN} & \text{else.} \end{cases} \quad (33)$$

Investigating the individual terms in detail, this expression can be "simplified" to:

$$a_k = \Pi_{c_\infty}^{doa} = \begin{cases} \text{PO}, & \text{if } 0 < \Pr(RUL_k \leq l) \cdot \Delta t \cdot (c_{\text{unav}} + c_{\text{inv}} - c_\infty) - c_{\text{inv}} \cdot \Delta t \\ & \quad + \Pr(l < RUL_k \leq l + \Delta t) \\ & \quad \cdot (c_{\text{unav}} + c_{\text{inv}} - c_\infty) \cdot (l + \Delta t - \\ & \quad \mathbb{E}[RUL_k \mid l < RUL_k \leq l + \Delta t]) \\ \text{DN} & \text{else.} \end{cases} \quad (34)$$

The full derivation of Eq. (34) is given in Appendix A.2.

As in (Kamariotis et al., 2024), we consider 3 approaches for



(a) Order at t_k, failure before delivery

(b) Order at t_k, failure after delivery

(c) Order at t_{k+1}, failure before delivery

(d) Order at t_{k+1}, failure after delivery

Figure 5. Exemplary incurred costs for each of the considered branches of the decision tree in Figure 4. The dotted black line indicates interruption of the component's function; hence, no costs are accumulated due to the continuation of the renewal-reward process.

choosing an appropriate initial guess $c_\infty \approx c_{\infty,0}$:

1. Without monitoring:
Considering a situation without monitoring (NM), renewal theory can be used to find the optimal time for ordering. The objective function for a deterministic ordering time t_{ord} is formulated as:

$$g^{NM}(t_{\text{ord}}) = \frac{1}{\mu_{T_F}}[(t_{\text{arr}} - \mathbb{E}[T_F \mid T_F \leq t_{\text{arr}}]) \cdot c_{\text{unav}} \cdot \Pr(T_F \leq t_{\text{arr}}) + (\mathbb{E}[T_F \mid T_F > t_{\text{arr}}] - t_{\text{arr}}) \cdot c_{\text{inv}} \cdot \Pr(T_F > t_{\text{arr}})], \quad (35)$$

where $t_{\text{arr}} = t_{\text{ord}} + l$. The corresponding initial estimate for the cost rate is found by solving the optimization problem:

$$c_{\infty,0} = \min_{t_{\text{ord}}} g(t_{\text{ord}}), \quad t_{\text{ord}} \in (0, \infty). \quad (36)$$

With monitoring and prognostics, the policy's cost rate will be lower than without. Hence, $c_{\infty,0}$ according to Eq. (36) is an upper bound on the true long-running maintenance cost per unit time. As a consequence, if this value of $c_{\infty,0}$ is used in the policy of Eq. (35), the penalty for early ordering is overestimated, which results in a PdM policy that prefers late ordering.

2. Perfect ordering:
An alternative approach to determining an initial estimate for the cost rate is to assume a perfect ordering policy, which could be obtained when the time of failure is known exactly, i.e., with perfect prognostics. The perfect ordering policy would be to order at $T_F - l$ if ordering were always possible. However, in our setting, one can only order at discrete decision times, so it must be determined if it is optimal to order at the nearest decision time before or after $T_F - l$. That depends on when exactly the component fails in this interval. Paying the inventory holding cost for almost the entire period Δt, can be worse than to pay the unavailability cost for only a fraction of Δt.
We derive $c_{\infty,0}$ with such a perfect policy by making a suitable set of assumptions detailed in Appendix A.3. The resulting analytical estimate for the long-running perfect ordering costs per unit time is

$$c_{\infty,0} = \frac{c_{\text{inv}} \cdot c_{\text{unav}}}{2(c_{\text{inv}} + c_{\text{unav}})} \frac{\Delta t}{\mu_{T_F}}. \quad (37)$$

The $c_{\infty,0}$ value obtained via Eq. (37) represents a lower bound to the cost rate achieved with the actual ordering policy; resulting in a more conservative ordering policy.

3. Intermediate value:
Any $c_{\infty,0}$ value between the upper bound (option 1) and lower bound (option 2) can be chosen.

The presented initialization options 1. to 3. can either be computed analytically with an analytical time-to-failure distribution or in a sample-based manner based on a set of run-to-failure trajectories.

Following step v, one can also regard c_∞ as a mere optimization parameter, and determine it's value via a data-based optimization with respect to the final policy performance $\hat{C}_\infty$, i.e., by solving Eq. (12).

4. NUMERICAL INVESTIGATIONS ON A VIRTUAL RUL SIMULATOR

We investigate the performance of the proposed heuristic policies in a controlled setting that enables accurate testing and verification. To this end, we make use of the RUL simulator introduced in (Kamariotis et al., 2024). The simulator provides *calibrated*[4] RUL predictions. This simulator enables the generation of an unlimited number of test data, which facilitates the exact evaluation of the different policies' performances.

We assume a Gaussian time-to-failure distribution of components: $T_F \sim \mathcal{N}(\mu = 225, \sigma = 40)$. The RUL simulator models prediction errors ϵ_k at time step t_k by a lognormal process $RUL_k = \epsilon_k \cdot RUL_{\text{true},k}$, where the logarithms of the prediction errors $\ln(\epsilon_k)$ follow a zero-mean multivariate normal distribution. This leads to marginal RUL prediction distributions equal to

$$\ln\left(RUL_k\right) \sim \mathcal{N}\left(\mu = \ln(RUL_{\text{true},k}) + \ln(\epsilon_k),\ \sigma = 0.4\right), \tag{38}$$

i.e., given an error sample $\epsilon_k^{(i)}$, f_{RUL_k} is the PDF of a lognormal distribution with mean $\mu_{RUL_k} \approx 1.083\epsilon_k^{(i)} \cdot RUL_{\text{true},k}$ and standard deviation $\sigma_{RUL_k} \approx 0.451\epsilon_k^{(i)} \cdot RUL_{\text{true},k}$. Thus, the simulator yields slightly biased RUL predictions, which is, however, not critical for performance. The RUL predictions at different t_k are correlated, with correlation length equal to $l_{\text{corr}} = 50$. A detailed description of the RUL simulator can be found in Appendix B. An example realization of RUL predictions is shown in Figure 1.

4.1. Replacement policy performance

We compare the performance of our proposed *doa* replacement policy with the benchmark heuristics $rh1$ and $rh2$ introduced in Section 3.2. For $rh1$ and *doa*, we include the default versions without parameter optimization as well as their optimized counterparts, denoted as $rh1^*$ and doa^*, respectively. The comparison is performed for $\Delta t = 10$ over a range of c_p/c_c ratios. The results are summarized in Figure 6. The performances of $rh1^*$ and doa^* are indistinguishable when graphed together with the other policies, which is why they are shown as one color.

One can see in Figure 6 that the default (unoptimized) version of our proposed *doa* heuristic achieves much lower cost ratios compared to the default versions of the benchmark policies. The reason why $rh1$ performs worse compared to *doa* is that it only considers the expected preventive/corrective costs directly attributable to the monitored component, and does not include the costs of the underlying renewal-reward process. $rh2$ improves upon this fallacy by including c_∞ in the objective function and solving for the optimal replacement time $r_{\text{rep}}^{*,rh2}$ at each decision time step. The final maintenance decision is taken based on whether $r_{\text{rep}}^{*,rh2}$ lies within the next decision time interval or not. However, this optimal replacement time is not available to the operator, but only the discrete decision times $t_k,\ t_{k+1}, \ldots$. Hence, having $r_{\text{rep}}^{*,rh2}$ does not necessarily help with taking the optimal maintenance decision regarding the discrete decision times, which ultimately leads to worse performance. By contrast, our proposed *doa* replacement heuristic specifically assesses the associated costs of the discrete replacement options at hand (or a one-step proxy thereof), and chooses the maintenance action accordingly. Importantly, the performance of the default *doa* is only marginally worse compared to the optimized policy. This is an indication that the proposed policy works even with a crude assumption on c_∞ and might be used even with no training data to optimize the policy (training data is of course still needed to train the RUL prediction).

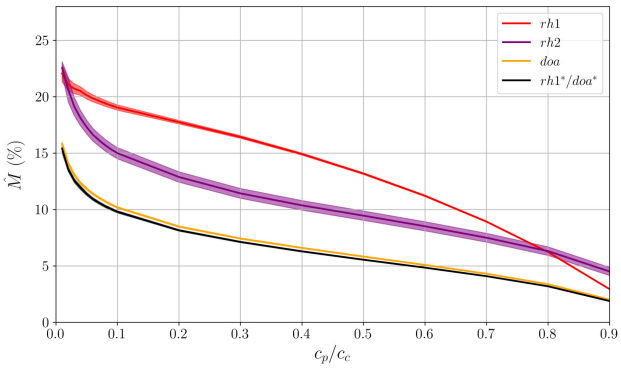


Figure 6. Comparison of our proposed *doa* replacement policy with the benchmarks $rh1$ and $rh2$. Since the optimized versions $rh1^*$ and doa^* yield the same performance, their performance is plotted as a single black curve. The following number of MC samples were used for the evaluation of each policy: $rh1 : 10^6$, $rh2 : 1.2 \cdot 10^4$, $doa : 10^6$. For the optimized policies, 10^6 MC samples were used for training. The figure shows the mean of each policy as a solid line, as well as the corresponding 95% credible intervals.

If one optimizes over the free parameter of each replacement heuristic, i.e., p_{thres} for $rh1$, or c_∞ for *doa*, the performances

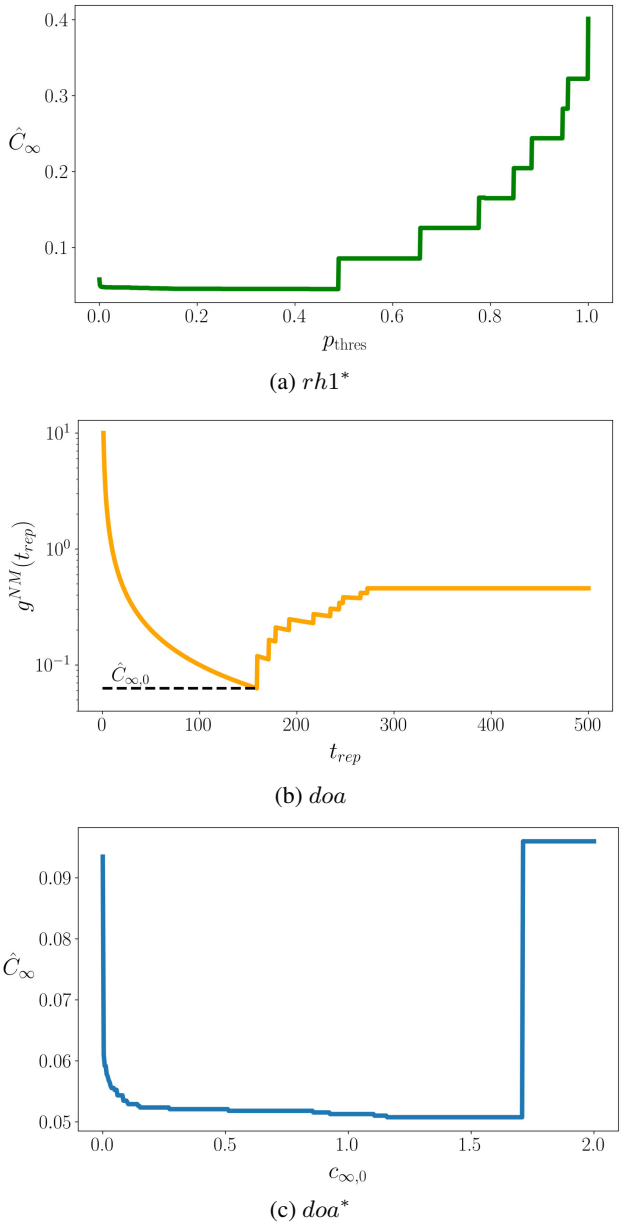
[4] A well-calibrated probabilistic model outputs predictions whose credible intervals correspond to the observed frequencies in the long run (Gneiting, Balabdaoui, & Raftery, 2007), i.e., the true RUL lies in the $(1-\alpha)\%$ credible interval in $(1-\alpha)\%$ of cases.

of the optimized policies converge to the same best performance. This indicates that it is possible to compensate an initially "bad" heuristic with optimization. However, these results are based on optimizing with a very large number of training data, specifically 10^6 run-to-failure samples. In practice, such large amounts of data are not available, especially for safety-critical systems. Thus, we investigate the effect of training data size on the performances of the respective heuristics.

In Figure 7, we examine the optimization for replacement heuristics $rh1^*$, doa and, doa^* in the case where only 10 sample histories are available. $rh1^*$ and doa^* perform the optimization directly over the policy performance ($\hat{C}_\infty$), whereas for doa we perform an optimization over $g(t_{\text{rep}})$ described in Eq. (16) for finding the cost rate estimate $c_{\infty,0}$, which is then subsequently used in the final policy. The differences between these two approaches have substantial consequences. For low amounts of failure samples, the objective function for $rh1^*$ and doa^* is constant over a wide range of p_{thres} values; see the plateaus in Figures 7a and 7c. The locations of the vertical jumps, as well as the final $\hat{C}_\infty$ values, depend on the specific set of samples. Because of these plateaus, there is no unique solution to the optimization problem. We dealt with this by generating uniform samples over the lowest plateau. But clearly, this is an issue when optimizing a heuristic with respect to the policy performance with limited data and impacts the accuracy of the solution.

In addition, finding the lowest plateau might be nontrivial, as gradient-based optimizers cannot be used and instead $\hat{C}_\infty$ has to be evaluated over the whole range of possible p_{thres} values. For the one parameter in our simulation setting, this is not a problem, but this could pose some issues for more parameters and more expensive policy evaluations. By contrast, with doa, the objective function for the parameter optimization for an initial cost rate estimate is better behaved. Although there are local minima where the optimizer can get stuck, the function is monotonically decreasing between jumps, which helps the optimizer. In our experience, even with these local minima, the optimizer usually finds the global minimum over the whole range of $\frac{c_p}{c_c}$ values. One can even avoid the optimization problem by choosing another initialization option for $c_{\infty,0}$, as presented in (Kamariotis et al., 2024); here, one simply uses $\frac{c_p}{\hat{\mu}_{T_F}}$, where $\hat{\mu}_{T_F}$ is the mean failure time computed over the set of available training samples. We have found that with this second initialization option, the heuristic's performance is slightly worse compared to the first option, but the results are almost indistinguishable.

Ultimately, the key question is how the optimized policies perform with limited training samples. Therefore, we test the performance of $rh1$ and doa for various training sample sizes for a fixed cost ratio $\frac{c_p}{c_c} = 0.1$. The results for 20 independent runs are shown as boxplots in Figure 8. They demonstrate that if one optimizes p_{thres} or c_∞ with only small amounts of training data, the resulting policy can have substantially worse performance than with the default parameter values $p_{\text{thres}} = \frac{c_p}{c_c}$ and $c_{\infty,0}$. This is classical overfitting. One can also see that our proposed doa heuristic is very robust against overfitting, and yields excellent results even in low data regimes.



(a) $rh1^*$

(b) doa

(c) doa^*

Figure 7. One example of the objective functions in the optimization of the heuristic parameters when only 10 failure samples are available. The cost ratio here is $\frac{c_p}{c_c} = 0.1$. The objective function for $rh1^*$ (top) and doa^* (bottom right) is the final policy performance ($\hat{C}_\infty$), whereas for doa it is the initial cost ratio estimate ($c_{\infty,0}$).

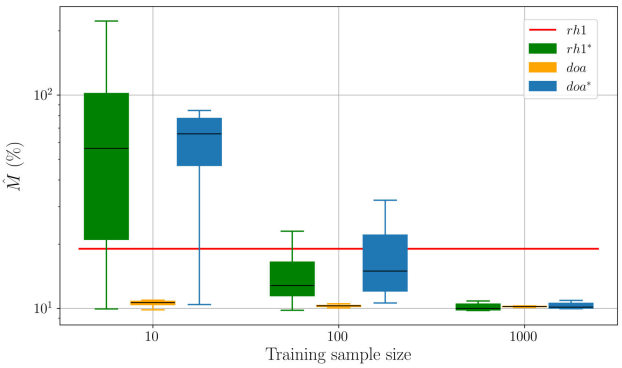


Figure 8. Boxplot comparison of our proposed doa policy with benchmark heuristics $rh1$ and $rh1^*$ for $c_p/c_c = 0.1$. For the optimization problem, training sample sizes of [10, 100, 1000] were used; evaluation of the final policies was performed with 10^6 samples, and the investigation was repeated 20 times, with the respective medians highlighted in black.

4.2. Ordering policy performance

We perform a similar investigation as in Section 4.1 to compare our proposed doa preventive ordering heuristic with the simple threshold policy $oh1$. We again include for both heuristics the default versions (without parameter optimization) as well as their optimized counterparts, denoted as $oh1^*$ and doa^*, respectively. The default threshold for $oh1$ is chosen as $p_{\text{thres}} = c_{\text{inv}}/c_{\text{unav}}$. The results for time step size $\Delta t = 10$ and ordering lead time $l = 11$ in function of the cost ratio $c_{\text{inv}}/c_{\text{unav}}$ are depicted in Figure 9. They are quite similar to the results of the preventive replacement setting in Figure 6, except that the differences between the policies are now much larger (in the order of hundreds of percent), which indicates a more difficult problem.

Again, the default version of our proposed doa ordering heuristic achieves significantly lower cost ratios compared to the default version of $oh1$ over most of the investigated range. In the range $c_{\text{inv}}/c_{\text{unav}} \in [0.62, 92]$, $oh1$ with $p_{\text{thres}} = c_{\text{inv}}/c_{\text{unav}}$ achieves better performance than our proposed method. This better performance is a coincidence rather than a systematic benefit and happens because for a cost ratio of $c_{\text{inv}}/c_{\text{unav}} = 0.8$, the optimal threshold is also close to 0.8. Furthermore, typical decision settings are located on the left side of the graph, i.e., for lower cost ratios, where the doa heuristic is superior.

Optimizing over the free parameter in each decision heuristic (p_{thres} for $oh1$ and c_∞ for doa), i.e., data-driven compensation of the process cost assumptions, results in the best performances. The curves of both optimized policies are visually indistinguishable.

Similarly to Section 4.1, we also investigate the robustness of these heuristics in the low data regime; the results of which are depicted in Figure 10. They show that doa without optimization is very robust against overfitting, even in low data regimes. However, due to the larger performance gap, also noticeable in Figure 9, the optimized $oh1^*$ policy, achieves a better performance in most cases already for 100 samples and a better median performance already for 10 samples. Overall, the best-performing policy is doa^*. While it is less robust than doa, it does have significantly lower $\hat{M}$ and is more robust than the $oh1^*$ policy. Nevertheless, in the 10-sample regime, one might also opt for doa.

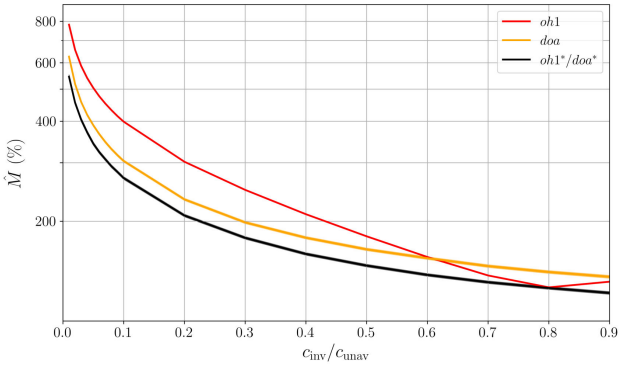


Figure 9. Comparison of our proposed doa ordering policy (Eq. (34)) with the simple heuristic ordering policy (Eq. (26)). Since the optimized version of both policies yields the same performance, they are represented as a single black line. The $oh1$ and doa policies were evaluated with 10^6 and 10^5 MC samples, respectively. The figure shows the mean of each policy as a solid line, as well as the corresponding 95% credible intervals.

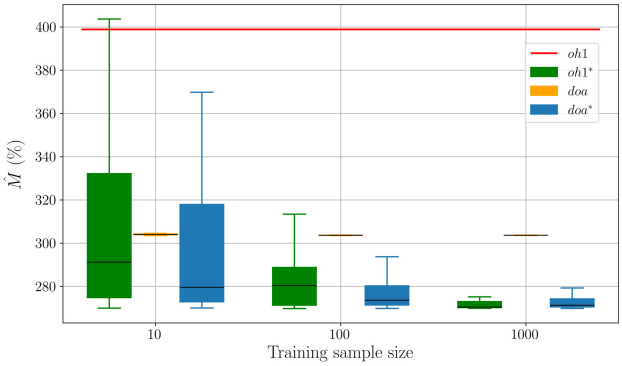


Figure 10. Boxplot comparison of our proposed doa policy with $oh1$ with optimized p_{thres} for $c_{\text{inv}}/c_{\text{unav}} = 0.1$. For the optimization problem, training sample sizes of [10, 100, 1000] were used; evaluation of all final policies was performed with 10^5 samples, and the investigation was repeated 40 times, with the respective medians highlighted in black.

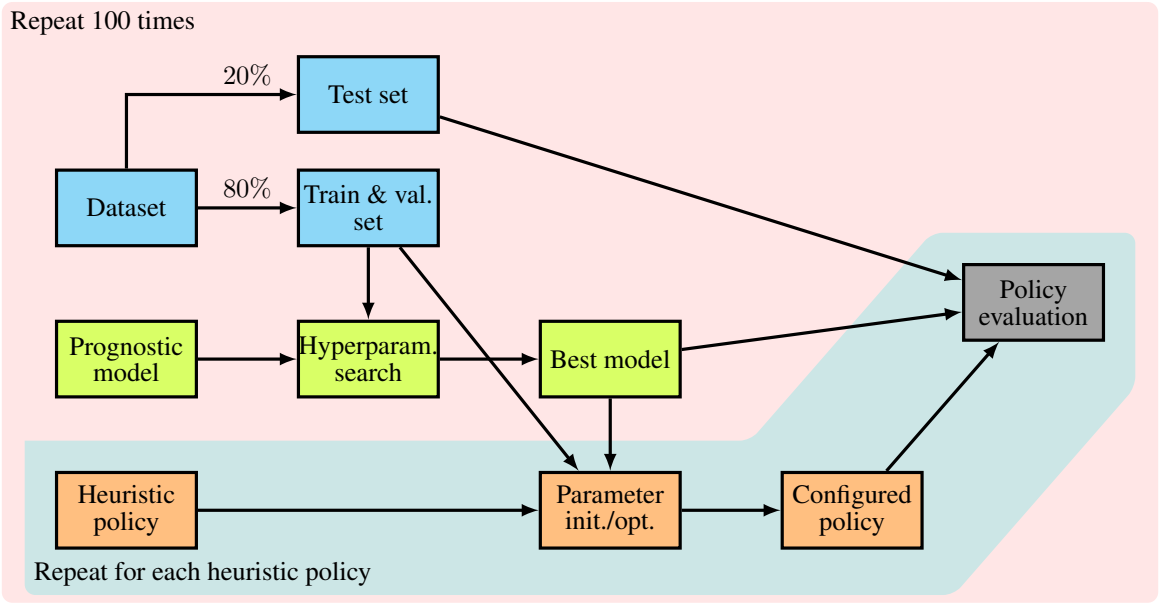


Figure 11. Evaluation procedure for comparing different heuristic policies.

5. CASE STUDY: PREVENTIVE ORDERING OF TURBOFAN ENGINES

In this section, we investigate our proposed method in a more realistic PdM setting. We utilize the C-MAPSS dataset[5], a well-known PHM benchmark problem concerning the degradation simulation of turbofan engines under different flight conditions (Saxena, Goebel, Simon, & Eklund, 2008).

The data of interest consists of 24 time series, comprising 21 sensor readings and 3 engine operation mode variables. We focus on the training set of FD001, one out of four data subsets, which contains run-to-failure trajectories of different engines under varying initial conditions, degradation parameters, and noise levels. The FD001 training set contains a total of 100 run-to-failure trajectories, which we use for our investigations. For a more detailed description of the dataset, the reader is referred to (Saxena et al., 2008).

For our investigation, we split the original training set into a training set and a test set. The training set is used to train a prognostic model as well as to perform potential initialization or optimization of a decision heuristic's free parameter. We suspect that this double use of the training set exacerbates the overfitting issue observed in Figures 8 and 10.[6] The final performance of the total PdM bundle, consisting of the prognostic model and the decision heuristic, is then evaluated on the test set. To show the variability of the whole training pipeline, we perform repeated random sub-sampling, which is also known as Monte Carlo cross-validation (MCCV). The full evaluation process is depicted in Figure 11.

As a prognostic model, we choose a Long Short-Term Memory (LSTM)-based neural network (NN), which is a standard choice for time series prediction tasks. The specific architecture follows the RUL classifier introduced in (Nguyen & Medjaher, 2019), i.e., the NN outputs a class label of the estimated RUL with a corresponding probability. For our ordering setting, the classes are chosen as i) $RUL \leq l$, ii) $RUL \leq l + \Delta t$, and iii) $RUL > l + \Delta t$. From the individual class probabilities, the full RUL-PDF required for the doa heuristic can be obtained by fitting a certain distribution type: for this work, following (Kamariotis et al., 2024), the lognormal distribution is chosen. A more detailed description of the employed architecture, the configuration, as well as the hyperparameter setup is given in Appendix C.

We focus the investigation on the ordering decision setting, because the prognostics algorithm achieves such a good performance for this data set that the replacement decision setting results in $\hat{M}$ values close to 0, even with a non-optimized heuristic, see (Kamariotis et al., 2024). With the presented evaluation pipeline, we compare the proposed doa/doa^* ordering heuristics with the benchmark policies $oh1/oh1^*$. The results are shown for multiple inventory to unavailability cost ratios in Figure 12. The box plots reflect the variability among the 100 MCCV repetitions. In each of these, we employ the random 80/20 train+validation/test split of the training FD001 dataset.

[5] C-MAPSS and other prominently used prognostics datasets are available at the NASA Prognostics Center of Excellence Data Set Repository: https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/

[6] Note that this "double use" of the training set relates to a practical finite-sample overfitting effect observed when both the prognostic model training and the parameter optimization of the heuristic policy rely on the same limited training data, rather than a methodological shortcoming of our evaluation protocol.

The resulting performances show larger variability for all investigated policies and cost ratios than with the RUL simulator of Appendix B. We suspect that this results from the combined effect of multiple factors, in particular the small test data size and uncalibrated RUL predictions from the prognostic model. Note that the shown boxplots in Figure 12 are based on correlated samples obtained with MCCV; thus, the true performance variances are likely even larger.

We observe that our proposed doa heuristic yields the best or close to the best performance over all investigated $c_{\text{inv}}/c_{\text{unav}}$ values. $rh1$ consistently shows the worst median performance over all cost ratios. As expected, optimizing the decision heuristic on the limited training dataset of 80 trajectories leads to overfitting, which results in the increased performance variations of $rh1^*$ and doa^*. This phenomenon is amplified in regions with low cost ratios. Furthermore, $rh1^*$ and doa^* show similar behaviour with comparable median performances and performance variations across the investigated cost ratios.

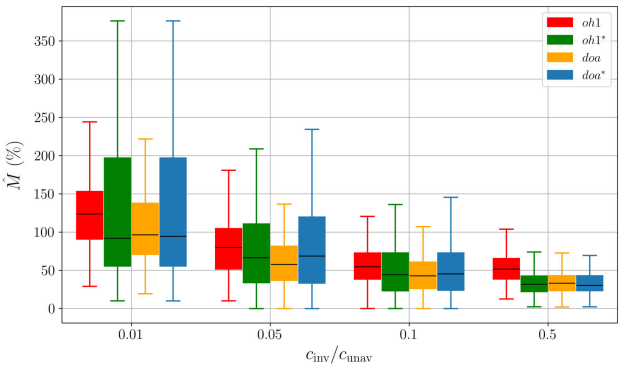


Figure 12. Boxplot comparison of our proposed doa/doa^* policies with benchmark heuristics $rh1/rh1^*$ for $c_p/c_c = [0.01, 0.05, 0.1, 0.5]$. 80 of the available 100 trajectories were used for training of the prognostic model and for initialization/optimization of the heuristics' free parameter; each policy's performance is evaluated on the remaining 20 trajectories. The investigation is repeated 100 times, with the respective medians highlighted in black.

6. DISCUSSION

This paper proposes a novel method to derive heuristic maintenance policies for PHM. We call these maintenance policies *heuristics* because they do not solve the full sequential decision optimization problem that is the maintenance process. We note that most currently existing maintenance policies proposed for PHM in the literature are of this kind. This is due to the limited data availability, which hinders the training of more sophisticated maintenance policies. An added advantage of heuristic policies is their explainability, which is important in practice.

In contrast to existing PHM maintenance policies, which were mostly derived without much theoretical justification, our proposed method does consider the full decision sequence explicitly. In order to end up with a heuristic policy, it simplifies the full sequential decision process by only considering and comparing the options of performing maintenance at the current and next opportunity. This online forward search with depth $d = 1$ is enhanced with a renewal theory-based offline estimation of the future costs (via the use of the long-running maintenance cost per unit time). Thus, combining online and offline approaches, our proposed method for deriving predictive maintenance policies falls in the category of *hybrid planning* (Kochenderfer et al., 2022).

The use of the one-step decision tree assumes a certain monotonicity of the underlying problem. To show this, let again denote $c_{M@t_k}$ as the expected costs of performing maintenance at time step t_k.[7] If one were to consider maintenance decision options also at all future time steps $t_{k+\tau},\ \tau > 1$, the optimal maintenance decision would correspond to the one derived from the one-step tree in all cases, except when $c_{M@t_{k+\tau}} < c_{M@t_k} < c_{M@t_{k+1}}$. In this case, the one-step tree yields the decision to maintain immediately, whereas the multi-step tree would yield the decision to do nothing. The one-step tree ignores that this case can occur, and thus assumes a monotonic cost structure of the maintenance actions over time. If decision-making is solely based on the latest RUL-PDF (i.e., no further information is extracted from the environment dynamics or from past RUL-PDFs), we believe that the one-step tree yields the optimal decision if the non-monotonic cost case does not occur.

Therefore, the one-step simplification of the decision process leads to policies that tend to perform a maintenance action too early. The representation of the costs of *maintain later* by $c_{M@t_{k+1}}$ undersells the advantages of waiting with a certain maintenance action. This results from the fact that one does not necessarily perform maintenance at t_{k+1}, but instead waits for more information and decides anew. The value of the option to perpetually wait and extend the life of the component for $2, 3, ..., n$ cycles is not included in the decision rule. Nevertheless, the achieved performances of the derived policies in Sections 4 and 5 show that the one-step assumption results in good policies for various cost and decision settings. Still, there is potential in further reducing the gap to the optimized policies, which correct for the conservatism of the proposed doa policies using data.

In future works, different approaches could be pursued to account for the value of the option to wait beyond the next decision time step. One approach could be to extend the one-step decision trees to include more future steps $t_{k+\tau},\ \tau > 1$ (i.e., forward search with a depth $d > 1$).[8] Another approach

[7] For example, for preventive replacement, $c_{M@t_k}$ and $c_{M@t_{k+1}}$ are given in Eqs. (18) and (21), respectively.

[8] A challenge here is that the assignment of the probabilities and associated expected costs of the subbranches would have to be made based on future

would be to keep the one-step formulation of *maintain now vs. maintain later*, but to use cost expressions that quantify the value of waiting and extending the life of the component for multiple time steps. Examples for such formulations could be the switch to age-based replacement (with the RUL-PDF as the time-to-failure PDF), or the assumption of replacement at the expected value (with $c_\infty \cdot \mathbb{E}[RUL]$ as expected cost savings).

The proposed policies are computationally more expensive than the simple threshold-based benchmark policies. The proposed policies require computation of consequences for each branch of the underlying tree, with the main cost drivers being the evaluation of the cumulative probabilities, $\Pr(a < RUL \leq b)$, and conditional expectations, $\mathbb{E}[RUL_k \mid a < RUL \leq b]$. They can either be computed analytically (as in this work) or, in the general case, via 1-D numerical integration. For the single-component decision settings of this paper, the respective decision trees have at most four branches. Comparing the policy expressions for replacement (Eqs. (13) and (23)) and ordering (Eqs. (26) and (34)), one can see that these cost-driving terms occur only once in the threshold-based policies, but $2-3$ times in the doa policies. Hence, evaluating the doa policies requires approximately 2–3 times the computation time, which is not critical. Accounting for the computational cost of the parameter optimization for $rh1^*/oh1^*$, the threshold-based policies are approximately 10–15 times more computationally expensive than the doa policies.

Optimized policies still yield the best performance when large amounts of training data are available. Interestingly, it does not seem to matter how the policy was originally formulated; optimization leads to the same final performance for these two relatively simple decision settings of preventive replacement and preventive ordering. Here, future work could explore the true performance limit by comparison with a partially observable Markov Decision Process based solution, assuming full knowledge of the environment (see, e.g., Kochenderfer, 2015).

Another advantage of parameter optimization is that it can also compensate for potential inadequacies of the prognostic model. Policies with initialized parameters, among them also the proposed doa policies, do not incorporate optimization and hence require a more reliable prognostic model to achieve a good maintenance performance.

The major drawback of optimized policies is their heavy overfitting in low data regimes, where one can end up with worse performance than with reference values or just simple age-based policies without monitoring. Here, our proposed doa policies with simple parameter initialization provide a remedy.

RUL-PDFs $f_{RUL_{k+\tau}}$, i.e., on information that only becomes available in the future. Correctly quantifying these probabilities and costs is nontrivial, but using the latest RUL-PDF f_{RUL_k} as an approximation of the true future distributions could be a first avenue to explore.

In both simulated settings as well as on the C-MAPSS dataset, our default proposed heuristics are robust against overfitting and thus can be reliably applied even in low data domains. Future work could investigate the application of the presented method to more complicated decision settings, e.g., ordering combined with replacement, or non-deterministic settings for $c_p, c_c, c_{\text{inv}}, c_{\text{unav}}, l$. Of particular interest is the formulation of heuristics for multi-component systems or fleets. An envisioned end result for practitioners could take the form of a look-up table, i.e., a guide on which heuristic to choose for the decision setting at hand. Note that for an M-component system, the number of branches in the doa decision tree grows as 2^{2M}. Hence, extending the doa policies to multi-component systems poses a greater computational challenge. We leave this for further research.

In this work, we have neglected discounting of future incurred costs, and used c_∞ to consider the costs of the underlying renewal-reward process in the doa decision rules (and $rh2$). For applications in which the overall lifetime of components is in the order of up to a few years, this is a reasonable assumption. However, to generalize the applicability to components with longer lifetimes, discounting should be considered. Van der Weide, Suyono, and Van Noortwijk (2008) generalize renewal-reward processes to include discounting with respect to any discount function that is non-increasing and monotonic over time. Here, one can formulate an expected equivalent average cost (EEAC) per unit time as a discounted version of c_∞. For instance, for exponential discounting described by $e^{-\rho t}$ with discount rate ρ over time t, EEAC (in the limit $t \to \infty$) can be obtained as (Van der Weide et al., 2008):

$$EEAC = \frac{\rho \mathbb{E}\left[e^{-\rho T_{\text{lc}}} C\right]}{1 - \mathbb{E}\left[e^{-\rho T_{\text{lc}}}\right]}. \tag{39}$$

Hence, to include discounting, one could potentially use EEAC instead of c_∞ to account for the costs of the underlying renewal-reward process, though it remains to be verified whether the heuristic policies derived in this paper would require modification.

7. CONCLUDING REMARKS

We present a novel and general method of deriving heuristic policies for predictive maintenance. The policies are based on a reduced decision tree to model the available decision options and their consequences. The probability of reaching a certain branch is computed with the RUL-PDF obtained from a prognostic model. The expected costs of each action are then computed as the sum of the maintenance action and failure costs, which are directly attributable to the monitored component, as well as the costs of the underlying renewal-reward process. In this way, the approach also accounts for the gain from a potential life extension of a component.

We applied the proposed method to two predictive maintenance settings, namely preventive replacement and preventive ordering. We investigated the performance of our proposed doa policies, together with commonly used benchmarks, on a simulated dataset as well as the C-MAPSS dataset. Our results show that the doa policies achieve lower cost rates than the presented benchmarks. In the case of preventive replacement, our proposed heuristic is almost as good as an optimized heuristic. When a large corpus of training data is available, one can compensate a "bad" heuristic by parametrization of the decision policy, with subsequent optimization over the parameter(s) with respect to the policy's final performance. However, in the context of low training data availability, this leads to overfitting. Here, the proposed doa policies have shown to be robust against overfitting in all investigated decision settings and datasets. If one uses a parametrized version of our proposed heuristic, optimization still leads to overfitting, with more or less variance compared to benchmark policies, depending on the specific decision setting.

On the one hand, this work highlights the importance of data, and how large datasets can lead to great performance through optimization. On the other hand, this work also accentuates that a lot of performance can be gained by taking a step back and performing a more theory-driven investigation. This is essential for applications with a limited amount of run-to-failure data, which is the case for many application domains. The better and more robust policies resulting from a theory-driven approach constitute another step towards optimal use of the data at hand and, ultimately, the improvement of efficiency and safety of engineering systems.

ACKNOWLEDGEMENT

This work was supported by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) through the aviation research program LUFO VI-3 in the project BIG-ROHU.

NOMENCLATURE

t	Time
Δt	Decision interval
t_k	k-th decision time: $k \cdot \Delta t, k = 1, 2, ...$
r_k	Relative time $t - t_k$
T_F	Time to failure
μ_{T_F}	Mean of the TTF distribution
$f_{T_F}(t)$	TTF distribution
$f_{RUL_k}(r)$	Predicted RUL distribution at t_k
$\Pr(RUL_k \leq u)$	Predicted failure probability until time u: $\int_0^u f_{RUL_k}(r)\,\mathrm{d}r$
c_p, c_c	Preventive and corrective cost
c_{unav}, c_{inv}	Unavailability and inventory cost rate
l	Lead time
a_k	Maintenance action at time step t_k
$\Pi^h_{\mathbf{w}}$	Policy of heuristic h depending on parameter vector $\mathbf{w}$
DN, PR, PO	"Do nothing", "preventively replace", "preventively order"
rhi, ohi	i-th benchmark replacement and ordering heuristics
p_{thres}	Probability threshold
$p_{F,k}$	Probability of failure until the next decision time $\Pr(RUL_k \leq \Delta t)$
t_{rep}, t_{ord}	Age-based replacement and ordering times
$c_{M_1@t_k, M_2@t_{k+1}}$	Expected cost of choosing actions M_1 and M_2 at t_k and t_{k+1}, respectively.
c_∞	Long-running maintenance cost rate
$c_{\infty,0}$	Initial c_∞ estimate
$\hat{M}$	MC estimate of the performance deviation from the optimal policy

REFERENCES

Abbas, A. N., Chasparis, G., & Kelleher, J. D. (2022). Interpretable hidden markov model-based deep reinforcement learning hierarchical framework for predictive maintenance of turbofan engines. *arXiv preprint arXiv:2206.13433*.

Asgarpour, M., & Sørensen, J. D. (2018). Bayesian based Prognostic Model for Predictive Maintenance of Offshore Wind Farms. *International Journal of Prognostics and Health Management*, *9*(1).

Bellman, R. (1966). Dynamic Programming. *Science*, *153*(3731), 34–37.

Bismut, E., & Straub, D. (2021). Optimal Adaptive Inspection and Maintenance Planning for Deteriorating Structural Systems. *Reliability Engineering & System Safety*, *215*, 107891.

Cha, J. H., Finkelstein, M., & Levitin, G. (2017). On preventive maintenance of systems with lifetimes dependent on a random shock process. *Reliability Engineering & System Safety*, *168*, 90–97.

Chao, M. A., Kulkarni, C., Goebel, K., & Fink, O. (2022). Fusing physics-based and deep learning models for prognostics. *Reliability Engineering & System Safety*, *217*, 107961.

Chelbi, A., & Aït-Kadi, D. (2001). Spare provisioning strategy for preventively replaced systems subjected to random failure. *International Journal of Production Economics*, *74*(1-3), 183–189.

Chen, C., Shi, J., Lu, N., Zhu, Z. H., & Jiang, B. (2022). Data-driven predictive maintenance strategy considering the uncertainty in remaining useful life prediction. *Neurocomputing*, *494*, 79–88.

Chien, Y.-H., Sheu, S.-H., & Zhang, Z. G. (2012). Optimal maintenance policy for a system subject to damage in a

discrete time process. *Reliability Engineering & System Safety*, *103*, 1–10.

De Jonge, B., & Scarf, P. A. (2020). A review on maintenance optimization. *European Journal of Operational Research*, *285*(3), 805–824.

Dong, E., Zhan, X., Yan, H., Tan, S., Bai, Y., Wang, R., & Cheng, Z. (2025). A data-driven intelligent predictive maintenance decision framework for mechanical systems integrating transformer and kernel density estimation. *Computers & Industrial Engineering*, *201*, 110868.

Doob, J. L. (1948). Renewal Theory From the Point of View of the Theory of Probability. *Transactions of the American Mathematical Society*, *63*(3), 422–438.

Elwany, A. H., & Gebraeel, N. Z. (2008). Sensor-driven prognostic models for equipment replacement and spare parts inventory. *IIE Transactions*, *40*(7), 629–639.

Ethier, S. N., & Kurtz, T. G. (2009). *Markov Processes: Characterization and Convergence*. John Wiley & Sons.

Fauriat, W., & Zio, E. (2020). Optimization of an aperiodic sequential inspection and condition-based maintenance policy driven by value of information. *Reliability Engineering & System Safety*, *204*, 107133.

Finkelstein, M. (2015). On the optimal degree of imperfect repair. *Reliability Engineering & System Safety*, *138*, 54–58.

Gneiting, T., Balabdaoui, F., & Raftery, A. E. (2007). Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *69*(2), 243–268.

Grall, A., Dieulle, L., Bérenguer, C., & Roussignol, M. (2006). Asymptotic failure rate of a continuously monitored system. *Reliability Engineering & System Safety*, *91*(2), 126–130.

Grimmett, G., & Stirzaker, D. (2020). *Probability and Random Processes*. Oxford University Press.

Guo, C., Wang, W., Guo, B., & Si, X. (2013). A maintenance optimization model for mission-oriented systems based on Wiener degradation. *Reliability Engineering & System Safety*, *111*, 183–194.

He, R., Tian, Z., Wang, Y., Zuo, M., & Guo, Z. (2023). Condition-based maintenance optimization for multi-component systems considering prognostic information and degraded working efficiency. *Reliability Engineering & System Safety*, *234*, 109167.

Hendrycks, D., & Gimpel, K. (2016). Gaussian Error Linear Units (GELUs). *arXiv preprint arXiv:1606.08415*.

Hinderer, K., & Waldmann, K.-H. (1999). Approximate solution of Markov renewal programs with finite time horizon. *SIAM journal on control and optimization*, *37*(2), 502–520.

Hodkiewicz, M., Bonser, D., Chin, A., & While, L. (2018). Optimizing fixed-interval maintenance periods for mobile asset sub-systems operating in remote locations. In *Phm society european conference* (Vol. 4).

Hoffman, J. D., & Frankel, S. (2018). *Numerical Methods for Engineers and Scientists*. CRC press.

Jack, N., Iskandar, B. P., & Murthy, D. P. (2009). A repair–replace strategy based on usage rate for items sold with a two-dimensional warranty. *Reliability Engineering & System Safety*, *94*(2), 611–617.

Jiang, R. (2009). An accurate approximate solution of optimal sequential age replacement policy for a finite-time horizon. *Reliability Engineering & System Safety*, *94*(8), 1245–1250.

Jimenez-Roa, L. A., Simão, T. D., Bukhsh, Z., Tinga, T., Molegraaf, H., Jansen, N., & Stoelinga, M. (2024). Maintenance Strategies for Sewer Pipes with Multi-State Degradation and Deep Reinforcement Learning. *arXiv preprint arXiv:2407.12894*.

Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, *101*(1-2), 99–134.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, *4*, 237–285.

Kamariotis, A., Tatsis, K., Chatzi, E., Goebel, K., & Straub, D. (2024). A metric for assessing and optimizing data-driven prognostic algorithms for predictive maintenance. *Reliability Engineering & System Safety*, *242*, 109723.

Kim, N.-H., An, D., & Choi, J.-H. (2017). Prognostics and Health Management of Engineering Systems. *Switzerland: Springer International Publishing*.

Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.

Kochenderfer, M. J. (2015). *Decision Making Under Uncertainty: Theory and Application*. MIT press.

Kochenderfer, M. J., Wheeler, T. A., & Wray, K. H. (2022). *Algorithms for Decision Making*. MIT press.

Koutas, D., Bismut, E., & Straub, D. (2024). An investigation of belief-free DRL and MCTS for inspection and maintenance planning. *Journal of Infrastructure Preservation and Resilience*, *5*(1), 6.

Kuipers, L., & Niederreiter, H. (2012). *Uniform Distribution of Sequences*. Courier Corporation.

Lee, J., & Mitici, M. (2023). Deep reinforcement learning for predictive aircraft maintenance using probabilistic remaining-useful-life prognostics. *Reliability Engineering & System Safety*, *230*, 108908.

Lei, X., & Sandborn, P. A. (2016). PHM-Based Wind Turbine Maintenance Optimization Using Real Options. *Int J Progn Health Manag*, *7*(1), 1–14.

Luque, J., & Straub, D. (2019). Risk-based optimal inspection strategies for structural systems using dynamic Bayesian networks. *Structural Safety*, *76*, 68–80.

Mercier, S., & Castro, I. (2013). On the modelling of imperfect repairs for a continuously monitored gamma

wear process through age reduction. *Journal of Applied Probability*, *50*(4), 1057–1076.

Mitici, M., de Pater, I., Barros, A., & Zeng, Z. (2023). Dynamic predictive maintenance for multiple components using data-driven probabilistic RUL prognostics: The case of turbofan engines. *Reliability Engineering & System Safety*, *234*, 109199.

Nguyen, K. T., & Medjaher, K. (2019). A new dynamic predictive maintenance framework using deep learning for failure prognostics. *Reliability Engineering & System Safety*, *188*, 251–262.

Nielsen, J. S., & Sørensen, J. D. (2018). Computational framework for risk-based planning of inspections, maintenance and condition monitoring using discrete Bayesian networks. *Structure and Infrastructure Engineering*, *14*(8), 1082–1094.

Pecht, M. (2009). Prognostics and Health Management of Electronics. *Encyclopedia of Structural Health Monitoring*.

Pineau, J., Gordon, G., & Thrun, S. (2006). Anytime Point-Based Approximations for Large POMDPs. *Journal of Artificial Intelligence Research*, *27*, 335–380.

Prakash, G., Narasimhan, S., & Pandey, M. (2017). Condition Based Maintenance of Low Speed Rolling Element Bearings using Hidden Markov Model. *International Journal of Prognostics and Health Management*, *8*(1).

Raiffa, H., & Schlaifer, R. (2000). *Applied Statistical Decision Theory* (Vol. 78). John Wiley & Sons.

Ran, Y., Zhou, X., Lin, P., Wen, Y., & Deng, R. (2019). A Survey of Predictive Maintenance: Systems, Purposes and Approaches. *arXiv preprint arXiv:1912.07383*, 1–36.

Reddi, S. J., Kale, S., & Kumar, S. (2019). On the Convergence of Adam and Beyond. *arXiv preprint arXiv:1904.09237*.

Saassouh, B., Dieulle, L., & Grall, A. (2007). Online maintenance policy for a deteriorating system with random change of mode. *Reliability Engineering & System Safety*, *92*(12), 1677–1685.

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation. In *2008 international conference on prognostics and health management* (pp. 1–9).

Shukla, L. (2023, 11 16). *Bayesian Hyperparameter Optimization - A Primer*. Retrieved from `https://wandb.ai/wandb_fc/articles/reports/Bayesian-Hyperparameter-Optimization-A-Primer--Vmlldzo1NDQyNzcw` (Accessed 05-August-2025)

Si, X.-S., Wang, W., Hu, C.-H., & Zhou, D.-H. (2011). Remaining useful life estimation–A review on the statistical data driven approaches. *European Journal of Operational Research*, *213*(1), 1–14.

Sidibe, I., Khatab, A., Diallo, C., & Kassambara, A. (2017). Preventive maintenance optimization for a stochastically degrading system with a random initial age. *Reliability Engineering & System Safety*, *159*, 255–263.

Smith, W. L. (1958). Renewal Theory and Its Ramifications. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *20*(2), 243–284.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*(1), 1929–1958.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning: An Introduction* (Vol. 1(1)) (No. 1). MIT press Cambridge.

Van der Weide, J., Suyono, & Van Noortwijk, J. (2008). Renewal theory with exponential and hyperbolic discounting. *Probability in the Engineering and Informational Sciences*, *22*(1), 53–74.

Wang, L., Chen, Y., Zhao, X., & Xiang, J. (2024). Predictive Maintenance Scheduling for Aircraft Engines Based on Remaining Useful Life Prediction. *IEEE Internet of Things Journal*.

Wang, L., Zhu, Z., & Zhao, X. (2024). Dynamic predictive maintenance strategy for system remaining useful life prediction via deep learning ensemble method. *Reliability Engineering & System Safety*, *245*, 110012.

Wang, W., & Wang, H. (2015). Preventive replacement for systems with condition monitoring and additional manual inspections. *European Journal of Operational Research*, *247*(2), 459–471.

Wang, Y., Wang, G., Wu, Y., Zhang, G., & Wu, M. (2025). An uncertainty-aware deep learning ensemble approach for effective cutting tool predictive maintenance decision-making. *Measurement Science and Technology*, *36*(2), 026116.

Wang, Z., Hu, C., Wang, W., Kong, X., & Zhang, W. (2015). A prognostics-based spare part ordering and system replacement policy for a deteriorating system subjected to a random lead time. *International Journal of Production Research*, *53*(15), 4511–4527.

Wu, F., Niknam, S. A., & Kobza, J. E. (2015). A cost effective degradation-based maintenance strategy under imperfect repair. *Reliability Engineering & System Safety*, *144*, 234–243.

Yan, T., Lei, Y., Li, N., Si, X., Pintelon, L., & Dewil, R. (2022). Online joint replacement-order optimization driven by a nonlinear ensemble remaining useful life prediction method. *Mechanical Systems and Signal Processing*, *173*, 109053.

Zhuang, L., Xu, A., & Wang, X.-L. (2023). A prognostic driven predictive maintenance framework based on Bayesian deep learning. *Reliability Engineering & System Safety*, *234*, 109181.

A. DERIVATIONS

For the derivation of the final *doa* policies, we omit the subscript k for ease of notation.

A.1. Derivation of doa replacement heuristic

Starting with the decision criterion in Eq. (22), we can expand all individual terms:

$$\begin{aligned} \cancel{c_p} + \cancel{c_\infty \cdot r} < p_F \cdot c_c - p_F \cdot c_\infty \cdot \mathbb{E}\left[RUL \mid RUL < \Delta t\right] \\ + \cancel{p_F \cdot c_\infty \cdot r} + \cancel{c_p} + \cancel{c_\infty \cdot r} - c_\infty \cdot \Delta t \\ - p_F \cdot c_p - \cancel{p_F \cdot c_\infty \cdot r} + p_F \cdot c_\infty \cdot \Delta t, \end{aligned} \tag{40}$$

which yields through reordering

$$c_\infty \Delta t < p_F \left[c_c - c_p + c_\infty \left(\Delta t - \mathbb{E}\left[RUL \mid RUL < \Delta t\right]\right)\right]. \tag{41}$$

A.2. Derivation of doa ordering heuristic

Starting with the decision criterion in Eq. (33), we investigate all individual terms separately, to see which ones cancel out:

$$\begin{aligned} c_{Ord@t_k} = \Pr\left(RUL \le l\right) \cdot \left[\left(\overset{1}{l} - \mathbb{E}\left[RUL \mid \overset{2}{RUL} \le l\right]\right)\right. \\ \left. \cdot\, c_{\text{unav}} + \left(\overset{3}{r} - \overset{4}{l}\right) \cdot c_\infty\right] \\ + \Pr\left(RUL > l\right) \cdot \left[\left(\mathbb{E}\left[RUL \mid \overset{5}{RUL} > l\right] - \overset{6}{l}\right)\right. \\ \left. \cdot\, c_{\text{inv}} + \left(\overset{3}{r} - \mathbb{E}\left[RUL \mid \overset{7}{RUL} > l\right]\right) \cdot c_\infty\right] \end{aligned} \tag{42}$$

$$\begin{aligned} c_{Ord@t_{k+1}} = \Pr\left(RUL \le l + \Delta t\right) \cdot \left(\overset{1}{l} + \overset{8}{\Delta t}\right. \\ \left. - \mathbb{E}\left[RUL \mid \overset{2}{RUL} \le l + \Delta t\right]\right) \cdot c_{\text{unav}} \\ + \Pr\left(RUL \le l + \Delta t\right) \cdot \left(\overset{3}{r} - \overset{4}{l} - \overset{8}{\Delta t}\right) \cdot c_\infty \\ + \Pr\left(RUL > l + \Delta t\right) \cdot \left(-\overset{6}{l} - \overset{9}{\Delta t}\right. \\ \left. + \mathbb{E}\left[RUL \mid \overset{5}{RUL} > l + \Delta t\right]\right) \cdot c_{\text{inv}} \\ + \Pr\left(RUL > l + \Delta t\right) \left(\overset{3}{r}\right. \\ \left. - \mathbb{E}\left[RUL \mid \overset{4}{RUL} > l + \Delta t\right]\right) \cdot c_\infty. \end{aligned} \tag{43}$$

Expanding all the terms and comparing them against their counterparts leads to Table A.1.

Collecting all the terms on the left and right side of Table A.1 yields the condition:

$$\begin{aligned} 0 < \Pr\left(l < RUL \le l + \Delta t\right) \cdot \left(c_{\text{unav}} + c_{\text{inv}}\right) \\ \cdot \left(l - \mathbb{E}\left[RUL \mid l < RUL \le l + \Delta t\right]\right) \\ + \Pr\left(l < RUL \le l + \Delta t\right) \cdot c_\infty \\ \cdot \left(\mathbb{E}\left[RUL \mid l < RUL \le l + \Delta t\right] - l\right) \\ + \Pr\left(RUL \le l + \Delta t\right) \cdot \Delta t \cdot \left(c_{\text{unav}} + c_{\text{inv}} - c_\infty\right) \\ - c_{\text{inv}} \cdot \Delta t, \end{aligned} \tag{44}$$

which can be rearranged to

$$\begin{aligned} 0 < \Pr\left(l < RUL \le l + \Delta t\right) \cdot \left(c_{\text{unav}} + c_{\text{inv}}\right) \\ \cdot \left(l + \Delta t - \mathbb{E}\left[RUL \mid l < RUL \le l + \Delta t\right]\right) \\ + \Pr\left(l < RUL \le l + \Delta t\right) \cdot c_\infty \\ \cdot \left(\mathbb{E}\left[RUL \mid l < RUL \le l + \Delta t\right] - l - \Delta t\right) \\ + \Pr\left(RUL \le l\right) \cdot \Delta t \cdot \left(c_{\text{unav}} + c_{\text{inv}} - c_\infty\right) \\ - c_{\text{inv}} \cdot \Delta t. \end{aligned} \tag{45}$$

The terms can finally be simplified to the heuristic ordering condition

$$\begin{aligned} 0 < \Pr\left(l < RUL \le l + \Delta t\right) \cdot \left(c_{\text{unav}} + c_{\text{inv}} - c_\infty\right) \\ \cdot \left(l + \Delta t - \mathbb{E}\left[RUL \mid l < RUL \le l + \Delta t\right]\right) \\ + \Pr\left(RUL \le l\right) \cdot \Delta t \cdot \left(c_{\text{unav}} + c_{\text{inv}} - c_\infty\right) \\ - c_{\text{inv}} \cdot \Delta t. \end{aligned} \tag{46}$$

Table A.1. Cost comparisons for ordering at t_k vs. t_{k+1}, where the respective terms from Eqs. (42) and (43) are sorted by row. The table shows the cancellation of terms and the resulting simplification of the equations.

	Expected costs for ordering at t_k	Expected costs for ordering at t_{k+1}
1	$\Pr(\cancel{RUL \leq l}) \cdot l \cdot c_{\text{unav}}$	$\Pr(RUL \leq l + \Delta t) \cdot l \cdot c_{\text{unav}} = [\cancel{\Pr(RUL \leq l)} + \Pr(l < RUL \leq l + \Delta t)] \cdot l \cdot c_{\text{unav}}$
	$\longrightarrow \quad 0$	$\Pr(l < RUL \leq l + \Delta t) \cdot l \cdot c_{\text{unav}}$
6	$-\Pr(\cancel{RUL > l}) \cdot l \cdot c_{\text{inv}}$	$-\Pr(RUL > l + \Delta t) \cdot l \cdot c_{\text{inv}} = -[\cancel{\Pr(RUL > l)} - \Pr(l < RUL \leq l + \Delta t)] \cdot l \cdot c_{\text{inv}}$
	$\longrightarrow \quad 0$	$\Pr(l < RUL \leq l + \Delta t) \cdot l \cdot c_{\text{inv}}$
2	$-\Pr(RUL \leq l) \cdot \mathbb{E}[\cancel{RUL} \mid RUL \leq l] \cdot c_{\text{unav}}$	$-\Pr(RUL \leq l + \Delta t) \cdot \mathbb{E}[RUL \mid RUL \leq l + \Delta t] \cdot c_{\text{unav}} = -\int_0^{l+\Delta t} f_{RUL_k}(r) \cdot r\mathrm{d}r \cdot c_{\text{unav}}$ $= -\left[\cancel{\int_0^l f_{RUL_k}(r) \cdot r\mathrm{d}r} + \int_l^{l+\Delta t} f_{RUL_k}(r) \cdot r\mathrm{d}r\right] \cdot c_{\text{unav}}$
	$\longrightarrow \quad 0$	$-\Pr(l < RUL \leq l + \Delta t) \cdot \mathbb{E}[RUL \mid l < RUL \leq l + \Delta t] \cdot c_{\text{unav}}$
5	$\Pr(RUL > l) \cdot \mathbb{E}[\cancel{RUL} \mid RUL > l] \cdot c_{\text{inv}}$	$\Pr(RUL > l + \Delta t) \cdot \mathbb{E}[RUL \mid RUL > l + \Delta t] \cdot c_{\text{inv}} = \int_{l+\Delta t}^{\infty} f_{RUL_k}(r) \cdot r\mathrm{d}r \cdot c_{\text{inv}}$ $= \left[\cancel{\int_l^{\infty} f_{RUL_k}(r) \cdot r\mathrm{d}r} - \int_l^{l+\Delta t} f_{RUL_k}(r) \cdot r\mathrm{d}r\right] \cdot c_{\text{inv}}$
	$\longrightarrow \quad 0$	$-\Pr(l < RUL \leq l + \Delta t) \cdot \mathbb{E}[RUL \mid l < RUL \leq l + \Delta t] \cdot c_{\text{inv}}$
3	$[\Pr(RUL \leq l) + \Pr(RUL > l)] \cdot c_\infty \cdot r$ $= \cancel{c_\infty \cdot r}$	$[\Pr(RUL \leq l + \Delta t) + \Pr(RUL > l + \Delta t)] \cdot c_\infty \cdot r$ $= \cancel{c_\infty \cdot r}$
	$\longrightarrow \quad 0$	0
4	$-\Pr(RUL \leq l) \cdot c_\infty \cdot l$	$-\Pr(RUL \leq l + \Delta t) \cdot c_\infty \cdot l = -[\cancel{\Pr(RUL \leq l)} + \Pr(l < RUL \leq l + \Delta t)] \cdot c_\infty \cdot l$
	$\longrightarrow \quad 0$	$-\Pr(l < RUL \leq l + \Delta t) \cdot c_\infty \cdot l$
7	$-\Pr(RUL > l) \cdot \mathbb{E}[RUL \mid RUL > l] \cdot c_\infty$ $= -\cancel{\int_l^{\infty} f_{RUL_k}(r) \cdot r\mathrm{d}r} \cdot c_\infty$	$-\Pr(RUL > l + \Delta t) \cdot \mathbb{E}[RUL \mid RUL > l + \Delta t] \cdot c_\infty =$ $-\left[\cancel{\int_l^{\infty} f_{RUL_k}(r) \cdot r\mathrm{d}r} - \int_l^{l+\Delta t} f_{RUL_k}(r) \cdot r\mathrm{d}r\right] \cdot c_\infty$
	$\longrightarrow \quad 0$	$\Pr(l < RUL \leq l + \Delta t) \cdot \mathbb{E}[RUL \mid l < RUL \leq l + \Delta t] \cdot c_\infty$
8		$\Pr(RUL \leq l + \Delta t) \cdot \Delta t \cdot (c_{\text{unav}} - c_\infty)$
9		$-\Pr(RUL > l + \Delta t) \cdot \Delta t \cdot c_{\text{inv}}$

A.3. Derivation of initial cost rate estimate option 2 for doa ordering heuristic

Suppose $t_k < T_F^{(i)} < t_{k+1}$ for a certain component. The decision that the operator now faces is whether they should order earlier and pay the inventory cost, or order later and pay the unavailability cost. Since the goal is to minimize the cost, the operator should choose to pay for the inventory if

$$c_{\text{inv}} \cdot (T_F - t_k) < c_{\text{unav}} \cdot (t_{k+1} - T_F)\,. \tag{47}$$

By inserting $t_{k+1} = t_k + \Delta t$ and reordering, the operator should pay the inventory cost if

$$T_F < t_k + \frac{c_{\text{unav}}}{c_{\text{inv}} + c_{\text{unav}}} \cdot \Delta t. \tag{48}$$

Hence, the decision on which cost to pay boils down to how close the component fails to t_k relative to t_{k+1}. We are therefore only interested in the remaining failure time[9]

$$R_F = T_F - t_k, \tag{49}$$

which, when inserted into Eq. (48), yields

$$R_F < \frac{c_{\text{unav}}}{c_{\text{inv}} + c_{\text{unav}}} \cdot \Delta t. \tag{50}$$

Therefore, the expected costs can be calculated as an expectation over RUL:

$$\mathbb{E}\left[C_{\min}\right] = \int_0^{\Delta t} C(t) f_{R_F}(t) \mathrm{d}t, \tag{51}$$

where $C(t)$ defines the minimal cost function (i.e., whether we pay inventory or unavailability costs). For performing the computation in Eq. (51), one either needs the exact distribution for the remaining failure time, or one can resort to a sample-based computation.

We can approximate $R_F \sim \mathcal{U}[0, \Delta t]$. We checked this approximation for the RUL simulator, and it is quite accurate for Δt sufficiently smaller than μ_{T_F} (Kuipers & Niederreiter, 2012). With this, the expected costs can be calculated as:

$$\begin{aligned}
\mathbb{E}\left[C_{\min}\right] &= \int_0^{\Delta t} C(t) \frac{1}{\Delta t} \mathrm{d}t \\
&= \int_0^{\frac{c_{\text{unav}}}{c_{\text{inv}}+c_{\text{unav}}}\Delta t} t \cdot c_{\text{inv}} \frac{1}{\Delta t} \mathrm{d}t \\
&+ \int_{\frac{c_{\text{unav}}}{c_{\text{inv}}+c_{\text{unav}}}\Delta t}^{\Delta t} (\Delta t - t) \cdot c_{\text{unav}} \frac{1}{\Delta t} \mathrm{d}t \\
&= \int_0^{\frac{c_{\text{unav}}}{c_{\text{inv}}+c_{\text{unav}}}\Delta t} t \cdot c_{\text{inv}} \frac{1}{\Delta t} \mathrm{d}t \\
&+ \int_0^{\frac{c_{\text{inv}}}{c_{\text{inv}}+c_{\text{unav}}}\Delta t} t \cdot c_{\text{unav}} \frac{1}{\Delta t} \mathrm{d}t \\
&= \frac{c_{\text{inv}}}{2\Delta t} \cdot \frac{c_{\text{unav}}^2}{(c_{\text{inv}} + c_{\text{unav}})^2} \Delta t^2 \\
&+ \frac{c_{\text{unav}}}{2\Delta t} \cdot \frac{c_{\text{inv}}^2}{(c_{\text{inv}} + c_{\text{unav}})^2} \Delta t^2 \\
&= \frac{c_{\text{inv}} \cdot c_{\text{unav}} \Delta t}{2(c_{\text{inv}} + c_{\text{unav}})^2} (c_{\text{inv}} + c_{\text{unav}}) \\
&= \frac{c_{\text{inv}} \cdot c_{\text{unav}}}{2(c_{\text{inv}} + c_{\text{unav}})} \Delta t.
\end{aligned} \tag{52}$$

Finally, we obtain an analytical expression for an initial estimate of the policy cost rate under perfect ordering:

$$c_{\infty,0} = \frac{c_{\text{inv}} \cdot c_{\text{unav}}}{c_{\text{inv}} + c_{\text{unav}}} \frac{\Delta t}{\mu_{T_F}}. \tag{53}$$

B. VIRTUAL RUL SIMULATOR

This section is more or less an iteration of the RUL simulator description in (Kamariotis et al., 2024).

We assume an underlying normal distribution of the time-to-failure of the components, i.e., $T_F \sim \mathcal{N}(\mu = 225, \sigma = 40)$. At each step t_k, we make a prediction about the remaining useful life of the component, denoted as RUL_k. This prediction deviates from the true RUL, denoted $RUL_{\text{true},k} = T_F - t_k$, by a certain error ϵ_k. We define this prediction in a logarithmic scale as:

$$\ln\left(RUL_k\right) = \ln\left(RUL_{\text{true},k}\right) + \ln\left(\epsilon_k\right). \tag{54}$$

In this work, we assume a multivariate normal (MVN) distribution for the logarithmic prediction errors over time. Specifically,

$$[\ln(\epsilon_1), \ldots, \ln(\epsilon_n)] \sim \text{MVN}(\mathbf{0}, \mathbf{\Sigma}). \tag{55}$$

The mean of the prediction errors is 0, hence the prognostic model is designed to be unbiased in log-scale. On the other hand, the covariance matrix $\mathbf{\Sigma}$ (not to be confused with the

[9] This definition identical to the RUL in the prognostic setting. However, to avoid confusion, we use a different name for case of no monitoring.

strategy Σ in Section 3!) is constructed as

$$\mathbf{\Sigma} = \mathbf{D} \cdot \mathbf{R} \cdot \mathbf{D}, \tag{56}$$

where $\mathbf{D}$ is a diagonal matrix containing the standard deviation of the prediction errors $\sigma_{\ln(\epsilon_k)}$ on the main diagonal. For this work, we fix $\sigma_{\ln(\epsilon_k)} = 0.4 \; \forall k$. $\mathbf{R}$ is a correlation matrix. If the predictions of the prognostic model are assumed to be independent at t_k, then $\mathbf{R}$ reduces simply to a diagonal matrix with 1 on the main diagonal. In this work, the prognostic model's predictions are correlated as

$$\mathbf{R} = [\rho_{ij}], \qquad \rho_{ij} = \exp\left(-\frac{|t_i - t_j|}{l}\right), \tag{57}$$

where l is the hyperparameter defining the correlation length.

The steps to generate the RUL-PDFs at each time steps are then:

1. Draw a failure sample $T_F^{(i)}$ from the underlying time-to-failure distribution.
2. Compute the true RUL values at each time step via $RUL_{\text{true},k}^{(i)} = T_F^{(i)} - t_k$.
3. Sample an error vector $[\ln(\epsilon_1^{(i)}), \ldots, \ln(\epsilon_n^{(i)})]$ from the MVN distribution in Eq. (55).
4. Compute the mean value of the prediction via Eq. (54): $\mu_{\ln(RUL_k)}^{(i)} = \ln(RUL_{\text{true},k}^{(i)}) + \ln(\epsilon_k^{(i)})$.
5. The RUL-PDF is then a lognormal distribution $\ln\left(RUL_k^{(i)}\right) \sim \mathcal{N}\left(\mu_{\ln(RUL_k)}^{(i)}, \sigma_{\ln(\epsilon_k)}\right)$.

C. NEURAL NETWORK SPECIFICATIONS

The NN architecture used in this work is depicted in Figure C.1. It includes at least one LSTM layer, at least one dropout layer, one fully connected (FC) layer and a Gaussian error linear unit (GELU) activation function (Hendrycks & Gimpel, 2016). The 25-dimensional input consists of 21 sensor measurements, 3 operation modes, and 1 variable indicating the age of the engine, where each input is passed as a window of length 50 to the network.

The fixed setting choices for all runs were ADAM (Kingma & Ba, 2014) as optimizer with AMSGRAD (Reddi, Kale, & Kumar, 2019) included, 100 MCCV splits, 20 runs per hyperparameter search (HS) via Bayesian optimization (Shukla, 2023), reduce on plateau as the learning rate (LR) scheduler, and an input window size of 50. The fixed parameters with their corresponding values are summarized in Table C.1.

Table C.1. Fixed training setting choices for all hyperparameter searches.

Setting	Choice
Optimizer	ADAM
AMSGRAD	Included
# HS (# MCCV splits)	100
# runs per HS	20
HS type	Bayesian optimization
LR scheduler type	Reduce on plateau
Input window size	50

The parameters included in the HS for each MCCV split are the batch size, LSTM hidden size, dropout (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014) probability, initial LR, as well as the LR scheduler factor and patience. All parameters with their corresponding search spaces are summarized in Table C.2.

Table C.2. Distributions for each hyperparameter included in the search, where $\{\cdot\}$, $\mathcal{U}[\cdot]$, and $\mathcal{LU}[\cdot]$ represent the discrete uniform, uniform, and log-uniform distributions, respectively.

Parameter	Distribution
Batch size	$\{64, 128, 256\}$
LSTM hidden size	$\{64, 128\}$
# LSTM blocks	$\{1, 2\}$
Dropout probability	$\mathcal{U}[0.0, 0.3]$
Initial lr	$\mathcal{LU}[10^{-5}, 10^{-1}]$
LR scheduler factor	$\mathcal{L}[0.5, 0.9]$
LR scheduler patience	$\{3, 4, ..., 25\}$

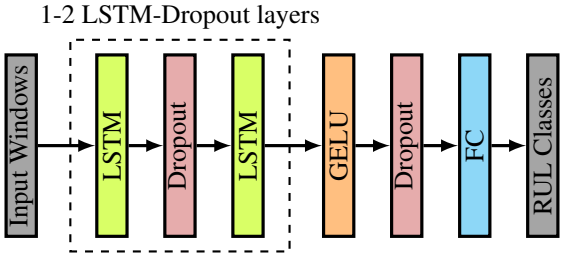


Figure C.1. NN RUL-classifier architecture with input and output (gray), lstm (lime), dropout (dark pink), and fully connected (cyan) layers, as well as GELU activation function (orange).

D. FURTHER INVESTIGATIONS OF RH2

D.1. Derivation

Here, we provide a derivation for $rh2$. For ease of notation, we again drop the subscript k wherever possible.

The overall goal is to find a maintenance policy with minimal c_∞. It is easier to work directly with costs instead of rates; hence, we fix the time frame, and then we can perform optimization over the costs. Consider time step t_k, where one decides whether to replace the component at some future time $t_k + r_{\text{rep}}$ or not. Since we base our cost calculations on the latest RUL-PDF, $f_{RUL}(r)$, the natural reference time frame is chosen as the largest residual time in the support of the distribution, i.e., $r_{\max} = \max\ r \mid f_{RUL}(r) > 0$.

Case 1: The component fails before replacement, i.e., $RUL \in [0,\ r_{\text{rep}}]$.

This incurs a corrective replacement cost at RUL and the long-running maintenance cost rate for the remaining time until $r_{\max}$:

$$C_c = c_c + c_\infty \cdot (r_{\max} - RUL). \tag{58}$$

Using the predicted RUL-PDF, we can calculate the expectation over C_c in the timeframe $[0,\ r_{\text{rep}}]$ as

$$\begin{aligned}\mathbb{E}\left[C_c\right] &= \int_0^{r_{\text{rep}}} \left[c_c + c_\infty \cdot (r_{\max} - r)\right] \cdot f_{RUL}(r)\mathrm{d}r \\ &= c_c \cdot p_F(r_{\text{rep}}) \\ &\quad + c_\infty \cdot \int_0^{r_{\text{rep}}} (r_{\max} - r) \cdot f_{RUL}(r)\mathrm{d}r,\end{aligned} \tag{59}$$

where $p_F(r_{\text{rep}}) = \int_0^{r_{\text{rep}}} f_{RUL}(r)\mathrm{d}r$.

Case 2: The component fails after the planned replacement, i.e., $RUL \in [r_{\text{rep}},\ r_{\max}]$.

This incurs a preventive replacement cost at r_{rep} and the long-running maintenance cost rate for the time frame $r_{\max} - r_{\text{rep}}$:

$$C_p = c_p + c_\infty \cdot (r_{\max} - r_{\text{rep}}). \tag{60}$$

Analogously to Eq. (59), we can calculate the expectation over C_p in the timeframe $[r_{\text{rep}},\ r_{\max}]$:

$$\begin{aligned}\mathrm{E}\left[C_p\right] &= \int_{r_{\text{rep}}}^{r_{\max}} \left[c_p + c_\infty \cdot (r_{\max} - r_{\text{rep}})\right] \cdot f_{RUL}(r)\mathrm{d}r \\ &= \int_{r_{\text{rep}}}^{r_{\max}} c_\infty (r_{\max} - r + r - r_{\text{rep}}) \cdot f_{RUL}(r)\mathrm{d}r \\ &\quad + c_p \cdot (1 - p_F(r_{\text{rep}})) \\ &= c_\infty \cdot \left[\int_{r_{\text{rep}}}^{r_{\max}} (r_{\max} - r) \cdot f_{RUL}(r)\mathrm{d}r \right. \\ &\quad \left. + \int_{r_{\text{rep}}}^{r_{\max}} (r - r_{\text{rep}}) \cdot f_{RUL}(r)\mathrm{d}r\right] \\ &\quad + c_p \cdot (1 - p_F(r_{\text{rep}})).\end{aligned} \tag{61}$$

The total expected cost of performing a preventive maintenance action at r_{rep} is then the sum of expected corrected and preventive maintenance costs in Eqs. (61) and (59), respectively:

$$\begin{aligned}\mathbb{E}\left[C_{PR@t_k+r_{\text{rep}}}\right] &= c_c \cdot p_F(r_{\text{rep}}) + c_p \cdot (1 - p_F(r_{\text{rep}})) \\ &\quad + c_\infty \cdot \left[\int_0^{r_{\text{rep}}} (r_{\max} - r) \cdot f_{RUL}(r)\mathrm{d}r \right. \\ &\quad + \int_{r_{\text{rep}}}^{r_{\max}} (r_{\max} - r) \cdot f_{RUL}(r)\mathrm{d}r \\ &\quad \left. + \int_{r_{\text{rep}}}^{r_{\max}} (r - r_{\text{rep}}) \cdot f_{RUL}(r)\mathrm{d}r\right] \\ &= c_c \cdot p_F(r_{\text{rep}}) + c_p \cdot (1 - p_F(r_{\text{rep}})) \\ &\quad + c_\infty \cdot \left[\int_0^{r_{\max}} (r_{\max} - r) \cdot f_{RUL}(r)\mathrm{d}r \right. \\ &\quad \left. + \int_{r_{\text{rep}}}^{r_{\max}} (r - r_{\text{rep}}) \cdot f_{RUL}(r)\mathrm{d}r\right].\end{aligned} \tag{62}$$

Since $\int_0^{r_{\max}} (r_{\max} - r) \cdot f_{RUL}(r)\mathrm{d}r$ is independent of r_{rep}, the objective function can be split into a part depending on r_{rep} and a constant part:

$$\begin{aligned}\mathbb{E}\left[C_{PR@t_k+r_{\text{rep}}}\right] = \mathbb{E}\left[C_{PR@t_k+r_{\text{rep}}}(r_{\text{rep}})\right] + \\ \mathbb{E}\left[C^{\text{const}}_{PR@t_k+r_{\text{rep}}}\right].\end{aligned} \tag{63}$$

For finding the optimal replacement time r^*_{rep}, the constant term is not of importance and thus can be dropped. The resulting formulation is identical to Eq. (15) when letting $r_{\max} \to \infty$, with the corresponding heuristic defined in Eq. (14).

D.2. Policy Extensions

For the use of $rh2$, one needs c_∞ to define the objective function. The accumulation of individual decisions then makes up the final maintenance policies, whose performance is again c_∞. Thus, to compute the optimal replacement time, one would already need to know the final performance of this policy. To circumvent this implicit definition, Kamariotis et al. (2024) propose three initial estimates. These estimates most likely do *not* coincide with the policy's final performance.

Hence, to align the assumed cost rate value with the actual performance, i.e., $\hat{C}_\infty = c_{\infty,0}$, we also investigate an iterative procedure as well as a direct optimization, which are outlined in Appendices D.2.1 and D.2.2.

D.2.1. Iterative procedure

The idea for the iterative procedure is to start with one of the $c_{\infty,0}$ options, use it for the objective function at each t_k, compute the resulting performance of the policy, $\hat{C}_\infty$, which is then used as the improved estimate $c_{\infty,1}$, and so on. Theoretically, if $rh2$ really represents the optimal policy, and the function $\hat{C}_\infty(c_{\infty,0})$ is convex/concave, then this fixed-point iteration scheme (Hoffman & Frankel, 2018) should converge to the globally optimal solution, at which $\hat{C}_\infty$, and consequently also $\hat{M}$, is minimal.

D.2.2. Direct optimization

Another data-driven option is to treat c_∞ as a mere parameter to be optimized. Similarly to p_{thres} of $rh1$, one solves an optimization problem to find the optimal value c^*_∞, for which $\hat{C}_\infty$ is minimal. If this value resulting from the optimization is substantially different from the converged value of the procedure in Appendix D.2.1, then $rh2$ does not accurately describe the costs arising from the underlying renewal-reward process, and there is room for improvement.

D.2.3. Results

The results for the comparison of the standard $rh2$ approach with the proposed extensions are depicted in Figure D.1. Surprisingly, the iteration scheme leads to a worse performance than by using just the default $c_{\infty,0}$ option 1. The reason for this is shown in Figure D.2. Although the iteration procedure converges to a value somewhere in between the bounds specified by initialization options 1 & 2, the optimal value, c^*_∞, for which the final policy performance $\hat{C}_\infty$ is minimal, is much larger than the upper bound. Moreover, the performance curve $rh2^*$ overlaps with the optimized policies $rh1^*/doa^*$ in Figure 6, i.e., one can again compensate a "bad" heuristic via data-driven optimization.

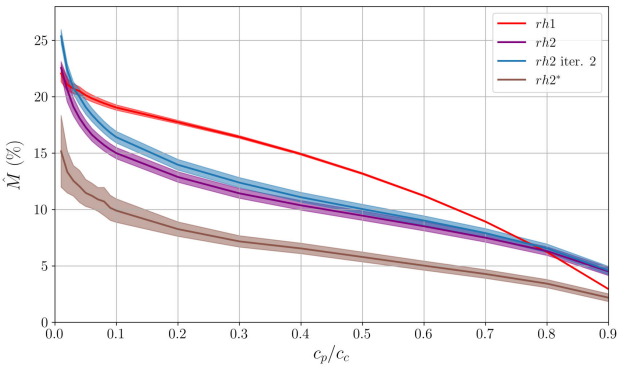


Figure D.1. Comparison of the default $rh2$ with its iterative and direct optimization variants. $rh1$ with the default probability threshold is also shown for reference. For all $rh2$ variants, $1.2 \cdot 10^4$ MC samples were used for training/evaluation. The figure shows the mean of each policy as a solid line, as well as the corresponding 95% credible intervals.

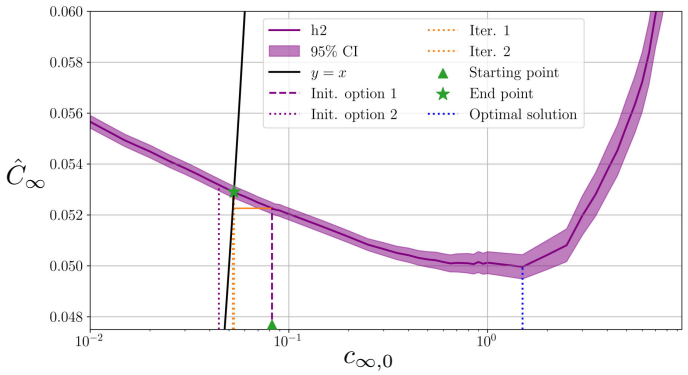


Figure D.2. $rh2$ performance with $\frac{c_p}{c_c} = 0.1$ plotted over a range of $c_{\infty,0}$ values, including two iterations as described in Appendix D.2.1, initialization options 1 & 2 from (Kamariotis et al., 2024), as well as the optimal value. The two iterations lie very close to each other, which is why they are visually overlapping to a single line.