ARTICLE IN PRESS
JID: EOR [m5G;February 8, 2022;11:37]

European Journal of Operational Research xxx (xxxx) xxx

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Production, Manufacturing, Transportation and Logistics

A dynamic “predict, then optimize” preventive maintenance approach using operational intervention data

Heletjé E. van Staden[a,*], Laurens Deprez[b], Robert N. Boute[c,d,e]

[a] University College Dublin, School of Business, Carysfort Ave, Carrysfort, Blackrock, Co. Dublin, Ireland
[b] University of Luxembourg, Luxembourg Centre for Logistics and Supply Chain Management, Faculty of Law, Economics and Finance, rue Richard Coudenhove-Kalergi 6, Luxembourg 1359, Luxembourg
[c] KU Leuven, Faculty of Economics and Business, Naamsestraat 69, Box 3555, Leuven 3000, Belgium
[d] Vlerick Business School, Technology and Operations Management, Vlamingenstraat 83, Leuven 3000, Belgium
[e] Flanders Make, VCCM, Gaston Geenslaan 8, Heverlee 3001, Belgium

ARTICLE INFO

Article history:
Received 27 July 2021
Accepted 24 January 2022
Available online xxx

Keywords:
Maintenance
Data-driven decision making
Markov decision process
Data pooling

ABSTRACT

We investigate whether historical machine failures and maintenance records may be used to derive future machine failure estimates and, in turn, prescribe advancements of scheduled preventive maintenance interventions. We model the problem using a sequential predict, then optimize approach. In our prescriptive optimization model, we use a finite horizon Markov decision process with a variable order Markov chain, in which the chain length varies depending on the time since the last preventive maintenance action was performed. The model therefore captures the dependency of a machine's failures on both recent failures as well as preventive maintenance actions, via our prediction model. We validate our model using an original equipment manufacturer data set and obtain policies that prescribe when to deviate from the planned periodic maintenance schedule. To improve our predictions for machine failure behavior with limited to no past data, we pool our data set over different machine classes by means of a Poisson generalized linear model. We find that our policies can supplement and improve on those currently applied by 5%, on average.

1. Introduction

Recent advances in data collection and algorithmic capabilities in data processing for decision making have resulted in renewed potential to streamline industry operations (Bastani, Zhang, & Zhang, 2020; Boute & Van Mieghem, 2021; Mišić & Perakis, 2019). Machine maintenance is an integral part of such industry operations of, for example, original equipment manufacturers (OEMs). Nonetheless, the most widely applied maintenance strategy used across industries today is still periodic maintenance. Under such a policy, preventive maintenance visits are based on a fixed schedule measured in calendar days or running hours. The schedule length may be estimated using the machine's believed failure rate, based on condition data or expert opinion, and is typically determined using a trade-off between expected preventive maintenance costs and (potential) failure costs. Periodic maintenance policies, however, lack predictive capability. As a result, the fixed maintenance interval often leads to maintenance being performed too early, i.e. before the end of a machine's useful life, or too late, i.e. after a costly failure. As such, a single periodic maintenance interval for different machines may be sub-optimal. Accordingly, condition-based and predictive maintenance strategies, based on observed machine condition information, promise significant cost savings (Coleman, Damodaran, Chandramouli, & Deuel, 2017; Haarman et al., 2018; Ivanov, Tang, Dolgui, Battini, & Das, 2020; Lueth, Patsioura, Williams, & Kermani, 2016; Manyika et al., 2015). However, the successful implementation thereof may be complex and generally requires large amounts of data and buy-in from end-users. As an alternative to measuring a machine's condition using sensor technology or human inspections, operational intervention data such as past failures and maintenance actions may instead be used to estimate machine failure probabilities and prescribe unscheduled preventive maintenance actions accordingly. Especially in the absence of (possibly expensive) condition monitoring information, these estimates may turn out to be valuable in an attempt to supplement scheduled periodic maintenance policies with customized predictions. However, work using such operational data is limited. We therefore investigate how generally available failure and main-

* Corresponding author.
E-mail addresses: heletje.vanstaden@ucd.ie (H.E. van Staden), laurens.deprez@uni.lu (L. Deprez), robert.boute@vlerick.com (R.N. Boute).

https://doi.org/10.1016/j.ejor.2022.01.037

tenance action data may be used to supplement and improve upon periodic maintenance. We do not propose to completely overhaul a company's existing periodic maintenance strategy (and thereby its related operations such as spare parts planning). Instead, we supplement such a strategy by providing insights into the likely future failures of a machine and recommend deviations from the existing maintenance schedule in a customized, structured manner. The resulting "predict, then optimize" approach is expected to reduce both waste (as a result of early maintenance) and failure costs (as a result of late maintenance).

Along with the challenges experienced in implementing predictive maintenance, we are inspired to use historical failure and maintenance data due to the fact that recent machine failures increase the likelihood of future failures (Deprez, Antonio, & Boute, 2020). When considering failure behavior, inherent heterogeneity between machines of different classes typically affects failure rates. In addition, machines are often run at different usage intensities, which further affects failure rates within and between machine classes. We therefore base our investigation on different machine classes and distinguish between three different usage intensities, defined as the proportion of available run time per year that a machine is used. The two classifications allow for investigating the effect of operating conditions, machine utilization in our case, and machine heterogeneity induced by machine classes on optimal maintenance policies.

We aim to minimize the expected maintenance costs of a machine over its maintenance contract period, using readily available operational intervention data, by focusing on two main objectives. Firstly, we investigate how to obtain maintenance prescriptions, defined as advising when to advance a scheduled periodic maintenance action and to perform unscheduled preventive maintenance. Secondly, we explore the value that can be extracted from such data and the resulting policy prescriptions for different machine class and usage intensity combinations. Due to the nature of the problem, we use a sequential predict, then optimize approach (Bastani et al., 2020; Mišić & Perakis, 2019). Our prediction model, in which we estimate conditional probabilities of future failures, is a combination of maximum likelihood frequency estimates, data pooling (Gupta & Kallus, 2020) and regression analysis (Ohlsson & Johansson, 2010). In our prescriptive maintenance (optimization) model, we focus on usage-based periodic maintenance. We assume that the maintenance interval is provided and use a Markov decision process (MDP) to optimize for unscheduled preventive maintenance actions that minimize maintenance costs. MDPs are a common tool to determine optimal maintenance policies. However, it is well known that Markov models of higher order tend to be over-fitted, with poor resulting performance (Buhlmann & Wyner, 1999). We therefore use a variable order state space to overcome the problem complexity of using historical maintenance data as input. Our approach is based on variable order Markov (VOM) chains, which allow for an improved trade-off between model bias and variance by varying the Markov chain length according to action and state realizations in the MDP.

Using the failures that occurred in a periodic maintenance interval to predict future machine failures via our prediction model, our prescriptive maintenance model therefore advises when to deviate from the normal periodic maintenance schedule and perform unscheduled preventive maintenance. The novelty of the model lies in the fact that we utilize data which are typically already captured by an organization and, as such, should be readily available to decision makers. Furthermore, we visualize the obtained policies from our prescriptive model in decision tree diagrams, thereby allowing for easy interpretation by decision makers and proactive adaptation of the maintenance schedule.

We validate our model with an OEM data set, where the OEM is responsible for maintaining their machines through maintenance contracts at a fixed, pre-determined yearly fee, referred to as full-service contracts. We find that the OEM can save, on average, 5% on maintenance costs when implementing our recommended policies. Our analysis highlights the importance of differentiating maintenance policies according to machine class and user behavior. Additionally, when compared to scheduled periodic maintenance only, our prescribed policies may be used to detect bad quality or poorly used machines that fail frequently and therefore advise preventive maintenance actions accordingly, potentially saving up to 44%. As such, an observed increase in failures serves as a proxy for poor running conditions or poor operation of a machine, for instance. When the latter is not explicitly available or measured by the decision maker, our approach is capable of capturing these 'hidden' features.

2. Literature review

Our research problem is related to two main research streams, namely (periodic) maintenance optimization and uncertainty modeling in maintenance. For a comprehensive review of maintenance optimization, we refer the interested reader to the recent work of De Jonge & Scarf (2020).

In the seminal work on periodic maintenance by Barlow & Hunter (1960), the authors determine structural results of optimal usage-based policies, given an associated failure distribution. They investigate policies that may either be rescheduled upon corrective maintenance or kept fixed regardless of the number of failures that occur between periodic maintenance actions. Recent advances in age-based periodic maintenance include optimizing the periodic maintenance interval in the absence of deterioration data, such as in Sgarbossa, Zennaro, Florian, & Calzavara (2020). The authors investigate the effect of the error of Weibull distribution parameter estimation on the resulting maintenance costs. In Drent, Kapodistria, & Boxma (2020), the authors investigate updating the maintenance interval based on the believed machine deterioration distribution after scheduled replacements by means of Bayesian learning, obtaining structural results for a computationally tractable myopic policy. Sanoubar, He, Maillart, & Prokopyev (2020) determine optimal age-based maintenance policies minimizing the average cost of maintenance for the case where preventive maintenance actions may deviate from the planned schedule. The authors show that the optimal policy minimizes the expected degree of deviation from the schedule and its variance. As is common in the literature, these papers assume a deterioration distribution over the machines for which periodic maintenance is optimized. However, a machine's failure distribution, and therefore its time-to-failure, may typically only be partially estimated based on available data or expert opinion.

A vast body of literature therefore focuses on estimating a machine's reliability. Regattieri, Manzini, & Battini (2010) perform a reliability analysis of parameter estimation for failing machines. The authors find that including censored data in the parameter estimation reduces errors in the reliability performance evaluation of the machines. Deprez et al. (2020) develop statistical failure prediction models based on risk factors such as machine age and intensity of use to improve total cost maintenance contract pricing strategies. Work considering uncertainty regarding the reliability of machine's condition data observations include Kim & Makis (2013), Van Staden & Boute (2021) and Dursun, Akay, & Van Houtum (2020). Kim & Makis (2013) and Van Staden & Boute (2021) determine optimal condition-based maintenance policies using condition observations partially related to a machine's actual deterioration. Dursun et al. (2020) determine optimal age-based periodic maintenance intervals given observations related to two possible underlying deterioration distributions. Condition-based maintenance generally improves on scheduled periodic maintenance only practices as they allow for dynamic (unscheduled) preventive maintenance based on the machine's observed condition.

Work based on condition data and which investigates postponing or advancing scheduled maintenance include Zhu, Peng, Timmermans, & van Houtum (2017), Poppe, Boute, & Lambrecht (2018), Van Oosterom, Elwany, Çelebi, & Van Houtum (2014) and Drent, Kapodistria, & Resing (2019). Zhu et al. (2017), Poppe et al. (2018) and Drent et al. (2019) assume knowledge of a component's deterioration rate to optimize maintenance of condition-based maintenance components with periodic maintenance components. They allow for unscheduled maintenance of the condition-based maintenance components at a periodic maintenance visit of the periodic maintenance components. In their work, Van Oosterom et al. (2014) allow for maintenance to be postponed until a defect is detected by means of condition data.

Despite advances in maintenance optimization using condition data, it may not always be possible to observe and correctly estimate a machine's actual deterioration rate using such data (Kim & Makis, 2013). In addition, it is not uncommon for condition data to only be obtainable at a cost (Van Staden & Boute, 2021). Given these restrictions, work considering operational intervention data, such as past failures and maintenance actions, as an alternative to optimize maintenance policies, is limited. In contrast to condition data, and since periodic maintenance policies are still the dominantly applied maintenance policy, operational data is typically easily available to decision makers at no additional cost or investment. As a result, we believe there is a gap in understanding the value of such operational data to improve on existing periodic maintenance practices. We therefore add to the literature on maintenance optimization using operational data in a predict, then optimize modeling approach. The approach provides predictive dynamics to scheduled periodic maintenance only strategies. In doing so, it partially counters the drawbacks of periodic maintenance where a fixed maintenance interval often leads to suboptimal maintenance actions that are either too early or too late, incurring unnecessary costs. We additionally make no assumptions regarding the underlying deterioration distribution of a machine, allowing for easy generalization of our approach to other machines.

3. Problem description, data and methodology

Our research is inspired by an OEM tasked with performing maintenance on an installed base of different machine types through full-service maintenance contracts. A full-service maintenance contract covers all preventive and corrective maintenance costs over its duration. The OEM therefore has an incentive to optimize its maintenance policy, prescribing when preventive maintenance should be performed, so as to minimize the contract maintenance costs. At present, the OEM bases maintenance of its installed base largely on a periodic, usage-based, maintenance strategy whereby a machine is maintained periodically after a predetermined, fixed number of running hours. However, the machines typically exhibit different failure rates due to, for instance, differences according to machine type induced heterogeneity. The OEM does not have access to machine condition information for these machines. The OEM does, however, keep maintenance records of past visits.

We investigate whether it is possible to improve on the OEMs current maintenance practices through the use of dynamic preventive maintenance policies via failure predictions obtained from the available operational intervention data. The problem is therefore twofold. Firstly, we investigate how to extract failure probability estimates from the data, making no assumptions about the underlying failure distributions of a machine being maintained, using a prediction model. Secondly, we use the estimates to determine when it is cost optimal to deviate from the periodic maintenance schedule and perform unscheduled preventive maintenance by means of a maintenance prescription model. The result is a sequential predict, then optimize approach. As highlighted in Section 2, work using operational data, as opposed to condition data, to inform dynamic maintenance practices is limited. Our work therefore provides valuable insights into the use thereof and extends the literature on maintenance optimization under uncertainty.

We have analyzed an operational data set consisting of various maintenance visits registered on about 4 000 full-service maintenance contracts, covering about 3 000 unique machines from different machine classes during the period 2004–2020 (we disguise the exact number for confidentiality reasons). The data set reveals that the OEM often deviates from the planned schedule. There is no apparent, clear reason for such deviations, but they are likely due to dynamic rescheduling of maintenance upon unrecorded inspection, unavailable maintenance staff or spare parts and, possibly, missing or incorrectly recorded data. A dynamic preventive maintenance policy, as proposed in this paper, may therefore be able to provide some structure to the existing dynamic nature of the OEMs current policy.

For our purposes, the machines in the data set are divided into seven classes. Two different types of technology and three levels of power output lead to six of these classes. In addition to the six machine classes, we combine data for the remaining machine types, which are not often sold, into a separate machine class. We label the clustered machines as machine class Type 4. Besides allowing for confidentiality preservation, this Type 4 classification allows us to validate customizing maintenance according to machine class, performed in Section 4. We provide an extensive description of the OEM data set in Appendix B and note that the application may easily be generalized to different machines (and machine portfolios) for which such data is available.

We note from the data that, both across and within machine classes, there exists significant variation in the usage intensity, i.e. the yearly average running hours, for different machines. We illustrate this spread in usage intensity in Appendix C, Figs. C.11 and C.12. Typically, increasing utilization increases failure frequency, see also Deprez et al. (2020). We therefore include usage intensity in our model and split such usage into three categories, respectively low, medium and high. We base the different categories on yearly running hour capacity specifications from the OEM, disguising the exact amount for confidentiality reasons. We let a low intensity machine be a machine operated at up to 50% total yearly running hour capacity. For medium and high intensity machines, we consider average running hours in a year of between 50% – 75% and greater than 75% total yearly running hour capacity, respectively. The categories allow for investigating the effect of operating conditions, machine utilization in our case, on the optimal policies. We control for the value of such a classification, by disregarding usage intensity, in Section 4.2.

Our objective is to prescribe dynamic preventive maintenance interventions to minimize the expected total maintenance costs over the maintenance contract length. To do so, we use a machine's usage, failure and maintenance history, i.e. the operational intervention data, to determine the probability of its future failure(s). Specifically, this probability of a future failure(s) is conditional on the number of failures that occurred in each decision epoch since the last performed preventive maintenance action (scheduled or unscheduled). Such failures are recorded in the prescription model's state space. Using the estimates, we determine when to deviate from the planned periodic maintenance schedule and perform unscheduled preventive maintenance, effectively advancing scheduled maintenance visits, at discrete time points referred to as decision epochs.

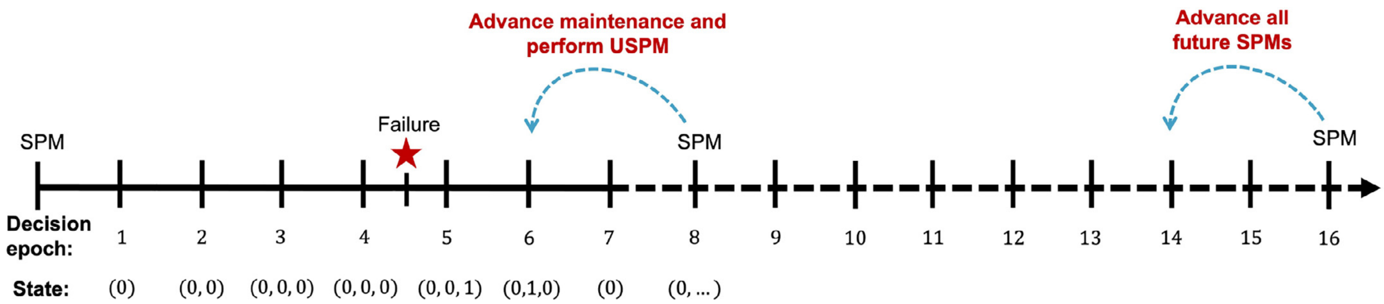


Fig. 1. An illustrative example of the problem investigated. Using past state information, in this case historical failures and maintenance actions for a specific machine class and usage intensity combination, we predict the probability of future failures. We use these predictions to recommend advancement of preventive maintenance as an unscheduled maintenance action. Should preventive maintenance be advanced, all future scheduled preventive maintenance actions are rescheduled accordingly.

We illustrate our approach in Fig. 1. We let T be the number of decision epochs in a periodic maintenance schedule (i.e. the number of decision epochs that elapse between two scheduled periodic maintenance actions), equal to 8 decision epochs in Fig. 1. We let $t = \{T, T-1, T-2, \dots, 1\}$ be the number of decision epochs remaining before the next scheduled periodic maintenance. Both scheduled (when $t = 0$) as well as unscheduled preventive maintenance (when $t > 0$ and $t \leq T-1$) resets the periodic maintenance schedule such that, at the beginning of the next decision epoch, the number of epochs left to the next scheduled periodic maintenance is $T-1$. An unscheduled preventive maintenance action is illustrated in Fig. 1 at decision epoch 6, where $t = 2$ satisfies both $t > 0$ and $t \leq T-1$. The next scheduled periodic maintenance action (and all future scheduled actions not illustrated), is automatically rescheduled accordingly to preserve the periodic maintenance interval length.

In the figure, a state of (0) means that no failures occurred in the previous decision epoch. Similarly, a state of (0,1,0) means that, considering the three most recent decision epochs, one failure occurred two decision epochs prior. The state complexity therefore increases over the machine's run time. In the figure, we limit the state space to looking back for a maximum of three decision epochs.

A key to obtain prescriptions for when to perform unscheduled preventive maintenance is the estimation of the transition probability matrices. A transition probability is defined as the probability of at least one failure in the next decision epoch, given the current failure history and action taken, i.e. the state space in Fig. 1. In order to determine these, a decision maker may have access to a data set consisting of a large number of observations. However, the number of parameters to be estimated may also be large, a challenge referred to as the small-data problem (Mišić & Perakis, 2019). As a result, despite large data sets, sufficient data to accurately estimate the different parameters may seldom be available (Kumwilaisak & Kuo, 2002). Determining our transition probabilities is an example of such a challenge since, for each machine class and usage intensity combination, a longer failure history, i.e. more 'look back' epochs, leads to a(n exponentially) larger state space. As the state space increases, so too does the number of transition probabilities to be estimated. In other words, the number of possible transitions increases exponentially in the number of epochs that we 'look back'. In turn, as the number of transition probabilities to be estimated increases, so too does the need for data observations, giving rise to the small-data problem.

Our approach therefore demands sufficient data for each failure state over every decision epoch that we look back, some of which may be limited or even unobserved for some or all machine classes. In order to avoid model over-fitting arising from insufficient data, data from different sources may be pooled to form a larger data set (Gupta & Kallus, 2020). To overcome the small-data problem, we transform our entire failure data set such that the failures of one machine class represent the failures of another machine class. This transformation enables us to determine the transition probability matrices for every machine class and usage intensity combination from the pooled data over all the machine classes by counting the failure state transitions from one decision epoch to the next.

3.1. The prescriptive maintenance model

We model the problem of determining when to perform unscheduled preventive maintenance, so as to minimize maintenance costs over the contract length, as a discrete-time finite horizon total reward MDP (Puterman, 1994). For our problem, we set the MDP time horizon equal to the maintenance contract length, in decision epochs, and denote it as N. The model therefore includes two different time units, namely T, the scheduled periodic maintenance interval length, and N, the maintenance contract length. An MDP is defined by a set of states, a set of actions, a transition function, denoting the transition probabilities between states, and a reward function. The MDP elements specific to our approach are described as follows.

State space We define L as the maximum order of our Markov chain (also known as finite memory). To capture the failure probability between preventive maintenance actions, we let $0 < L < T$ be the maximum number of decision epochs we allow the model to look back. We refer to L as the maximum look-back period, or number of epochs, for failure prediction. For each machine class, we let the MDP state space be the combination of failures per decision epoch for, at most, the most recent L epochs. We define a failure vector, consisting of the number of failures in the last $l = \min\{T-t, L\}$ decision epochs, as $\mathbf{s} = [s_{n-l}, s_{n-(l-1)}, \dots, s_{n-1}]$ where $s \in S = \{0, 1, 2, 3, \dots\}$ is the number of failures that occurred during decision epoch $n = \{1, 2, \dots, N\}$. We note that decision epoch n refers to the absolute time unit within the contract period, whereas t refers to the current decision epoch (time unit) within the scheduled maintenance interval.

Action set We allow for unscheduled preventive maintenance visits in the periodic maintenance interval, based on the probability that a machine will fail in the next decision epoch given the machine's recent failure history. For $t > 0$, two actions are therefore available to the decision maker. These are either to continue normal machine operations without intervention ($a =$ NPM), or to perform an unscheduled preventive maintenance action ($a =$ UPM). When action UPM is chosen, unscheduled preventive maintenance is performed and the maintenance schedule is reset to T epochs remaining until the next scheduled maintenance action. When $t = 0$, scheduled periodic maintenance is performed. For modeling purposes, we let $a =$ SPM when $t = 0$. Preventive maintenance, both SPM and UPM, is performed at the beginning of a decision epoch. We define the action set as $A = \{\text{SPM}, \text{UPM}, \text{NPM}\}$.

Reward function The reward function consists of the different maintenance visit costs. Preventive maintenance costs are incurred at the beginning of a decision epoch. The decision maker pays a surcharge for performing an unscheduled preventive maintenance action due to, for instance, expediting spare parts and freeing up technician time. We let the cost of preventive maintenance, $a \in \{\text{UPM, SPM}\}$, be $C_P(a)$. Upon machine failure during a decision epoch, corrective maintenance is performed. Corrective maintenance costs, $C_F(s)$, are based on the number of failures, s, that occurred during epoch n. Corrective maintenance is more expensive than preventive maintenance, such that $C_P(\text{SPM}) \le C_P(\text{UPM}) \le C_F(s)$ for $s \in S \setminus \{0\}$.

Transition probability matrices Depending on the action, a, taken and the time, t, in the periodic maintenance schedule, the machine state transitions between the failure states according to a transition probability matrix. We derive the transition probability matrices from the observed failure history from our data set. Preventive maintenance, either scheduled or unscheduled, is imperfect and restores the machine to an improved deterioration state according to the transition probability matrix for preventive maintenance (UPM or SPM). Since both actions UPM and SPM perform preventive maintenance, where UPM is unscheduled at $t > 0$ and SPM is scheduled at $t = 0$, we assume that both actions are identical in restoring the machine to an improved deterioration state. We therefore use the same transition probability matrix for the two preventive maintenance actions. For an epoch in which preventive maintenance is performed, we allow the resulting transition probability matrices to depend on the number of failures in the immediate preceding decision epoch, defined as $p_t^a(s|\mathbf{s})$ where $\mathbf{s} = [s_{n-1}]$ for $a \in \{\text{UPM, SPM}\}$.

The transition probability for action NPM, to continue normal machine operations without intervention, depends on the failure behavior in each decision epoch since the last periodic maintenance. For NPM, with $t > 0$ epochs remaining in the current periodic maintenance interval, we define $p_t^{NPM}(s|\mathbf{s})$ as the probability that the machine will fail s times by the beginning of the next decision epoch given that the current failure history over the past l ($\min\{T-t, L\}$) decision epochs is $\mathbf{s}$. The transition probability matrix for NPM therefore depends on the position, t, in the maintenance interval, T, and is determined as follows. For $t = T-1$, preventive maintenance was performed in the immediate preceding decision epoch. In the transition probability matrix, for $L \ge 1$, we thus only consider the failure behavior of one preceding decision epoch, the epoch in which preventive maintenance was performed at the start of the epoch. For $t = T-2$, preventive maintenance was performed two decision epochs prior. For $L \ge 2$, we thus consider the failure behavior of the two preceding decision epochs in determining the transition probability matrix. We continue similarly until $T - t > L$. For all remaining decision epochs in the current interval, we consider the failure behavior of the L preceding decision epochs in determining the transition probability matrices. We discuss the failure probability estimation approach which we use to populate the transition probability matrices based on the data set, in detail in Section 3.2.

The definition of the MDP allows us to minimize the total maintenance costs over the average contract duration, N, by formulating the problem as a finite horizon total cost MDP. We let the value function, $V_n(t, \mathbf{s})$ (Puterman, 1994), be

$$V_n(t,\mathbf{s}) = \begin{cases} V_N(t), & \text{if } n = N, \\ H_n(0,\mathbf{s},\text{SPM}), & \text{if } n < N \text{ and } t = 0, \\ \min_{a\in A}\{H_n(t,\mathbf{s},a)\}, & \text{if } n < N \text{ and } t > 0, \end{cases} \tag{1}$$

where

$$V_N(t) = \begin{cases} C_P(\text{SPM}), & \text{if } t = 0, \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

and the function $H_n(t, \mathbf{s}, a)$ is given as

$$H_n(t,\mathbf{s},a) = C_P(a) + \sum_{i\in S\setminus\{0\}} p_t^a(i|\mathbf{s})C_F(i) + \sum_{j\in S} p_t^a(j|\mathbf{s})V_{n+1}(\mathcal{T}, \mathbf{s}\cup j \setminus \{s_{n-(T-1)}\}), \tag{3}$$

for

$$\mathcal{T} = \begin{cases} t-1, & \text{if } a = \text{NPM}, \\ T-1, & \text{if } a = \text{UPM, SPM}. \end{cases} \tag{4}$$

The first term on the right-hand side of Eq. (3) is the preventive maintenance costs incurred, depending on whether action UPM or SPM was chosen. When action NPM is chosen, no preventive maintenance costs are incurred, such that $C_P(\text{NPM}) = 0$. The second term denotes the expected corrective maintenance costs for epoch n. The final term denotes the expected cost-to-go for epoch n, calculated as the minimum future expected maintenance costs. From the value function in (1), we determine the optimal prescription policies. The policies consist of the set of actions which minimizes the expected maintenance costs given the possible failure state combinations for each decision epoch in a maintenance interval over the given horizon.

3.2. The data-driven failure prediction model

For every decision epoch in the periodic maintenance interval, we can observe a transition to one of the failure states, including zero failures. Recall that the failure vector $\mathbf{s} = [s_{n-L}, s_{n-(L-1)}, \ldots, s_{n-1}]$ captures the failure history of the machine for the last L decision epochs. The current decision epoch is n, T is the length of a periodic maintenance interval, $s = \{0, 1, \ldots, k\}$ and $s = k$ refers to an epoch in which k or more failures has occurred. To determine the transition probabilities for a decision epoch in which preventive maintenance is performed, we count the transitions between every failure state, in the immediate preceding decision epoch, to each failure state in the decision epoch in which preventive maintenance is performed. We obtain the probability of observing failure state zero (respectively one, two,...) in a decision epoch in which preventive maintenance is performed as follows. If the known failure state in the preceding decision epoch was zero (respectively one, two,...), we take the ratio of the total number of transitions from this failure state zero (respectively one, two,...) to failure state zero (respectively one, two,...) in the preventive maintenance epoch and the total transitions out of state zero (respectively one, two,...) in the preceding epoch. We similarly obtain the probability of observing the remaining failure states.

Given the small-data problem, a drawback of using such time dependent transition probability matrices is that the resulting probabilities may become less reliable the larger L becomes and the more failure states per epoch we consider in the model. We cope with this possible reduced reliability by pooling the data from each of the seven different machine classes, based on a weighted average approach, to obtain transformed data sets (one for each machine class and usage intensity combination), which we use as input for the prediction model. The weights are determined using generalized linear models (GLMs) and defined as the likelihood that a failure from one machine class represents a failure from another machine class. For example, considering a Type 1 machine, a weight of 0.5 for a Type 2 machine means that, approximately, for every two failure observations in the Type 2 data set, a Type 1 machine will fail once. To transform failures from a Type 2 machine class to representative failures of a Type 1 machine class, we therefore multiply the failure occurrence of Type 2 failures with the GLM weights for a Type 2 machine class. We repeat the process for each of the remaining five machine classes, and for each usage intensity, to obtain a transformed data set for the Type 1 machine class across usage intensities. We similarly apply this approach to all seven machine classes, obtaining different transformed data sets for each usage intensity where each data set consists of the same number of observations as the original failure data set.

While pooling data from different sources to improve prediction accuracy is not new (Gupta & Kallus, 2020), we show the robustness of our specific approach to the choice of weights in Appendix E. We evaluate the spread in expected costs based on the standard error obtained for each estimated weight. Based on the analysis, we find that, relative to the current policy observed from the data, the spread in expected costs is constant at approximately 2 – 3.5% for the weights and machine classes considered. The results indicate that our approach not only increases the failure data sample size for each machine class, it also increases the reliability of the transition probability matrices.

To determine the weights, we fit Poisson GLMs (Ohlsson & Johansson, 2010), with the machine class and usage intensity as explanatory variables, to the number of failures per decision epoch. In the GLM framework, the response variable, in our case the number of failures per decision epoch, follows a distribution from the exponential family of distributions, including Poisson, Bernoulli and gamma. The categorical variables, in our case the machine class and usage intensity, can be included as explanatory variables. We choose the Poisson distribution to model the number of failures per epoch as it is typically used to model counting data. The results of Fig. D.13 (see Appendix D) supports our choice of the Poisson distribution. The Poisson GLM for our purpose takes the following form,

$$E[N_{\text{fail},i}] = g^{-1}(\beta_0 + \beta_1 \cdot \text{machine\_class}_i + \beta_2 \cdot \text{usage\_intensity}_i), \tag{5}$$

where the expected number of failures $E[N_{\text{fail},i}]$ for a decision epoch i is modeled by an intercept β_0 and the influence of both the machine class β_1 and the usage intensity β_2. The link function, $g(\cdot)$, is set to $\log(\cdot)$. Since we observe different failure behavior in a decision epoch directly following a preventive maintenance action, we split the data set into epochs with and epochs without such maintenance and fit separate models to these subsets. The calibrated Poisson GLM can be used to calculate each failure state's observation probability during a decision epoch in which maintenance was performed, or not, for each machine class and usage intensity combination. For instance, the probability $P_{\text{PM}}(s|x)$ of s failures on machine class x in a preventive maintenance epoch can be obtained from the GLM. Comparing these probabilities for the different machine classes enables us to transform the data of machine class x to the data of machine class y. A transition for a maintenance epoch on machine class and usage intensity combination y can be transformed to a transition on machine class and usage intensity combination x by counting such a transition with weight,

$$w_{\text{PM}}(s)_{y,x} = \frac{P_{\text{PM}}(s|x)}{P_{\text{PM}}(s|y)}, \tag{6}$$

where s is the number failures observed at the end of this transition between epochs. Each transition is now assigned a machine class and usage intensity combination specific weight, as determined by the Poisson GLM. The approach allows us to use the entire data set over all machine classes and usage intensities to determine the transition probabilities for a single machine type and usage intensity combination.

Upon transforming the data using the Poisson GLM, there may still be failure states for which we have limited to no data, depending on the maximum order of our Markov chain L and (indirectly) the periodic maintenance interval. As L increases, the number of possible failure states increases along with the sample standard error over the failure states. The sample standard error is given as

$$\sigma = \sqrt{\frac{p_t^a(s|\mathbf{s})(1 - p_t^a(s|\mathbf{s}))}{m_{T-t}}},$$

where m_{T-t} is the count frequencies for $T - t$ decision epochs since the last preventive maintenance action was performed. We therefore set $L \geq 1$ equal to the largest decision epoch look-back value over all machine classes for which the sample error of a failure state first exceeds 5%. The 5% limit allows for a transition probability accuracy trade-off between sufficient data and the failure history.

In addition to our pooled data approach, a number of different failure prediction approaches, such as extrapolating missing failure observations for each machine class, may be investigated for benchmarking purposes. Initial results of such methods reveal that our pooled approach performs well in reducing prediction variance. In what follows, we will focus on whether maintenance records may be used to prescribe future maintenance actions, and therefore do not focus on further improving the model's prediction accuracy.

4. Case study: Data-driven application of the prescriptive maintenance model

We apply our approach to prescribe preventive maintenance interventions using the OEM data set with historical failures and preventive maintenance visits registered by about 4 000 maintenance contracts. We benchmark our prescriptive decision model results against maintenance practices currently implemented by the OEM, as observed from the data. We refer to this policy as the current policy. Additionally, we compare our results to a (Markov) process where only scheduled preventive maintenance (SPM) is performed at a fixed periodic maintenance interval (and unscheduled preventive maintenance is not allowed). We refer to this policy as the SPM policy. We refer to the results from our model as the USPM policy, to emphasize that we allow for unscheduled maintenance in our approach. To aid interpretation of our results, we provide a number of descriptive statistics in Appendix C. We optimize the MDP model of Section 3.1 using value iteration reviewed in Puterman (1994). We focus on our key results in this section. We therefore relegate an extensive discussion on determining the prescriptive maintenance model parameters and transition probability matrices for this MDP, given the data set, to respectively Appendix D and Appendix E.

We note that, for the sake of confidentiality, we adjusted the data and disguised some of the parameters. We are internally consistent with our use of the adjusted data.

4.1. Visualizing policy guidelines

Since our USPM model is history dependent based on the minimum of either the number of look-back periods, L, or the number of decision epochs that passed since preventive maintenance was last performed, we can visualize the optimal policy as a decision tree. Visualizing the policy as a decision tree promotes intuitive and easy decision making based on the failure history of the current maintenance interval. In addition, the trees can be used for both fast insights into the policy structure as well as predicting likely future states and actions, thereby assisting maintenance planning (and related activities, such as inventory decisions).

We illustrate the steady state optimal policy for a machine class in Fig. 2. In the figure, every level (labeled $t = \ldots$) represents a different decision epoch in the maintenance interval, starting with a preventive maintenance action, either scheduled (SPM action) or unscheduled (UPM action), as the root node and the next scheduled periodic maintenance action is t epochs later. Recall that t is

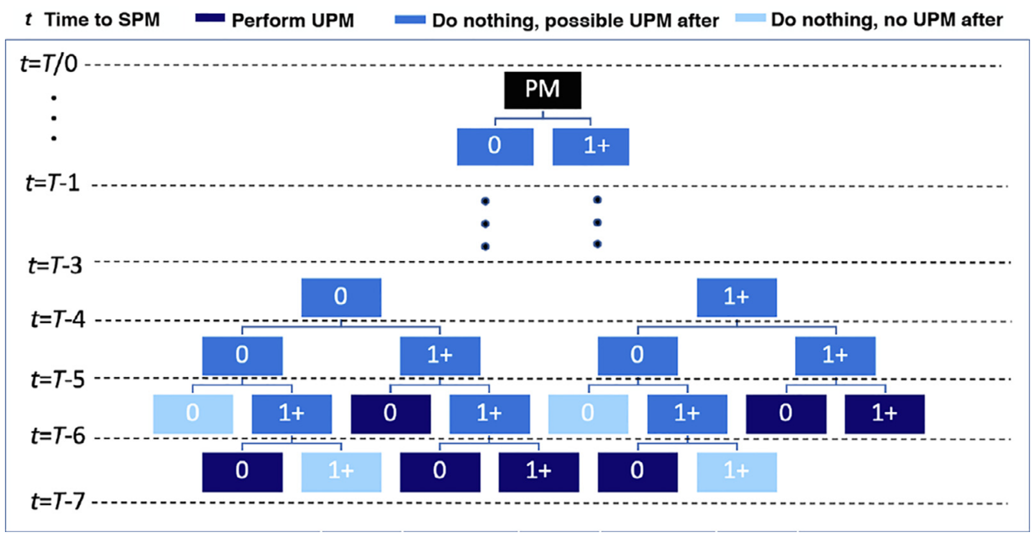


Fig. 2. The prescribed policy for a machine class. Light and dark blue nodes are terminal nodes, restarting the interval. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

defined as the number of decision epochs remaining until the next SPM action, with T defined as the number of decision epochs between SPM actions ($T = 8$ in Fig. 2). We use three different colors, light blue, medium blue and dark blue, to assist interpretation of the tree. Every dark blue terminal node represents a UPM action, after which the policy is restarted at the root node. Should the last level ($t = T - 7$ in Fig. 2) be reached and the current node does not recommend a UPM (a light blue node), an SPM is performed at the beginning of the next decision epoch and the policy is restarted at the root node.

A terminal node is represented by either light or dark blue. A light blue node represents a decision epoch for which preventive maintenance is not recommended (action NPM), after which no unscheduled preventive maintenance is recommended until the next scheduled preventive maintenance in the current periodic maintenance cycle. This recommendation of waiting until the next scheduled preventive maintenance regardless of the failures that may still occur, may be due to either a small chance of observing costly failures until the next preventive maintenance, or due to the fact that the combination of the number of costly past failures and the expected number of future costly failures outweigh the benefit expected to be gained from an unscheduled preventive maintenance. The use of light blue nodes allow us to significantly declutter the tree. A dark blue node represents a decision epoch for which action UPM is recommended, after which the periodic maintenance cycle is reset and the policy restarted at the root node. A medium blue node represents a non-terminal node for which action NPM is recommended. Unscheduled preventive maintenance may be recommended in future decision epochs, depending on the realized failures in future epochs in the current periodic maintenance cycle, such that the decision tree is continued to the next node.

While flexibility in performing maintenance when and as required is a key to successfully implementing preventive maintenance policies, it may not always be possible to perform immediate preventive maintenance due to, for instance, technician set-up time and spare part availability. Therefore, combining our model's prescribed policy, as represented in Fig. 2, with the failure probabilities of our transition probability matrices (implicit to Fig. 2, but not shown), may serve as a planning tool to identify the likeliest moment of a UPM action in the future given the current failure history. For instance, using the policy of Fig. 2 we know, for $t = T - 6$ and a recent failure history $\mathbf{s} = [0, 0, 1+]$, that if we continue with normal machine operations in the current epoch, the optimal policy recommends a UPM action at the beginning of the next decision epoch, $t = T - 7$, with probability $p_6^{\text{UPM}}(1+|\mathbf{s})$.

4.2. Benchmarking with current practice and scheduled periodic maintenance only policies

We next benchmark our model results (the USPM policy) against both the dynamic maintenance policy extracted from the OEM data set (the current policy) and the maintenance policy resulting from the periodic maintenance policy in which only scheduled preventive maintenance is allowed (the SPM policy). While benchmarking against the current policy, we highlight poor performing examples when one or both machine class and usage intensity classifications are disregarded. In doing so, we show the potential value of dynamic maintenance policies customized according to such classification.

Current policy To maintain the dynamic nature of the OEMs maintenance policy, we average the maintenance costs for each machine class over its different contracts for the current policy. We therefore do not assign a specific periodic maintenance interval to the current policy. For our USPM policy, we use the interval recommended by the OEM (see Appendix D for additional analysis).

We benchmark our USPM model against the current policy for each machine class and usage intensity. Without loss of generality, we focus on machine Types 2, 5 and 7 and plot the results in Fig. 3. From the figure, it may be seen that the average costs of the OEM policies follow a similar pattern to those obtained by our USPM model, suggesting that the OEM generally adapts its maintenance policy fairly well according to usage intensity (should the costs be extrapolated from running hours-based to calendar-based, the expected costs for a low intensity machine will naturally be lower than that for a higher usage intensity).

We find that our USPM policy generally achieves lower expected costs when compared to the OEM policy, with savings of up to 17%, for example, for the Type 2 low intensity machine combination. For a small number of cases, our USPM model is outperformed by the OEM. An example includes the high intensity machine Type 5 combination. We show in Fig. C.12 that the OEM generally increases the average periodic maintenance schedule (for all machine classes illustrated) as the usage intensity increases. This

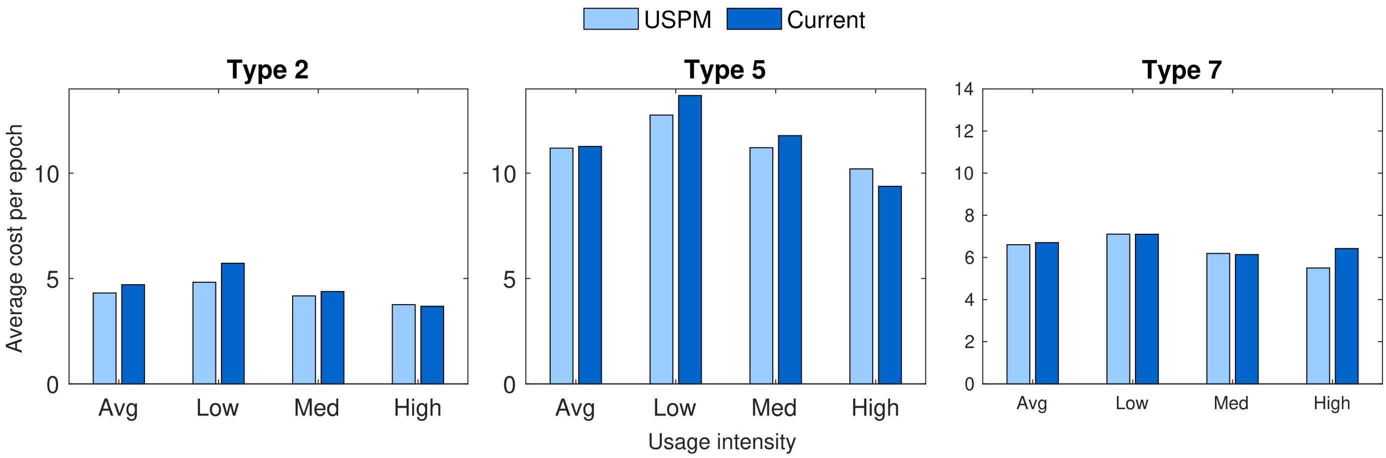


Fig. 3. Expected costs for machine class and usage intensity between our policy and that currently in use by the OEM.

behavior may be attributed to the fact that the ratio of failures and maintenance actions per running hour for machines which are idle for longer periods of time is typically higher than that for machines which are idle less often, as is the case for low intensity and high intensity machines respectively. As such, the optimal maintenance interval for higher usage intensity machines may be longer. We note that, when translating running hours into calendar time, the time interval would be shorter for high intensity machines. In Table C.4, we additionally show that over 40% of the OEMs Type 5 machines are run at high intensity. We may therefore conjecture that the OEM focuses their maintenance policy on high intensity machine usage for a Type 5 machine, likely with a maintenance interval longer than that used in our model (supported by Fig. C.12). Similar arguments may be made for the remaining cases in which the current policy outperforms our USPM policy. The results therefore suggest that our maintenance interval may be too short for some of the high intensity machine types. Nonetheless, we find that the OEM can improve upon its current maintenance practices allowing for maintenance decisions based on recent failures, as done in our USPM model. We note that some machine classes have limited data for low and high intensity usage, such as the high intensity Type 7 machine. For machine intensity classes with such limited data, our model takes advantage of the available data from all the machine classes through the transformation that results from our Poisson GLM. The transformation enables us to provide insights into the expected failure behavior of those usage intensity and machine class combinations for which we have limited actual data. As a result, we can recommend a customized maintenance policy, given this transformation, while limiting overfitting to small sample data.

We next highlight the increased benefit to the OEM, when considering both usage intensity and machine class classifications in determining their maintenance policy. In order to do so, we investigate the effect on expected costs when machine usage intensity is disregarded, such that maintenance decisions are made based on machine type only. The results, when compared to the current policy, are shown in Fig. 4. From the figure, we note that our model can improve on the expected costs of the current policy, when disregarding usage intensity, by up to 6.5%. For the Type 4 and 5 machine classes, however, our model fairs slightly worse.

Recall that machine class Type 4 consists of a mix of different, infrequently sold machines, and may therefore be likened to a policy that does not customize maintenance according to machine class. Given the poor performance of our model for a Type 4 machine class when compared to the current policy, it is likely that the OEM performs customized maintenance on the different machine types clustered together as Type 4. The results allow us to infer that a single maintenance policy for different machine classes may result in excessive maintenance costs. The Type 4 machine class results therefore highlight the value of customizing maintenance according to machine types, which we do for the remaining machine classes.

For the Type 5 machine class, we can attribute the decline in our model's performance, comparative to the current policy, to the large proportion of high intensity machines of Type 5 (Table C.4 and Fig. 3). The result indicates the potential downside of disregarding machine usage intensity in determining optimal maintenance policies. Given Figs. 3 and 4 and the related changes in policy performance, we conclude that, the larger the degree of heterogeneity in an installed base (such as larger differences in usage intensity or machine characteristics captured by machine classes), the higher the potential for maintenance cost savings when following maintenance policies optimized to such heterogeneity, as done in our USPM policy.

SPM policy Figure 5 shows the incremental value of our USPM policy over the SPM policy, disregarding usage intensity. Our policy generally outperforms the SPM policy by 0.2 - 0.8%, depending on the machine class. The main benefit of our model, however, is that it is able to identify when to deviate from the SPM policy. The value of our policy is therefore highest for bad quality or poorly used machines which, with a small probability, fail more often than the average machine.

For instance, using the policy from Fig. 2, we highlight two different possible sample paths in Fig. 6. We simulate both sample paths over one maintenance interval (eight decision epochs) for 100 000 runs and compare the results for a USPM and SPM policy in Fig. 7. The figure shows the value of the USPM model for machines of this class that fail often. From the figure, for 90% of the time, we expect approximately no savings for sample path 1. There is a small probability of saving up to 45% in one maintenance interval. Similarly, for sample path 2, we observe a small probability of saving up to 25%. Since performing a UPM is more expensive than an SPM, it is possible that the USPM policy may, at times, be more expensive than the SPM policy. From the figure, the probability of observing this cost increase is small. Based on the comparisons, our policies can assist the OEM to formalize their dynamic maintenance policies and, for the majority of usage intensity and machine class combinations, to further reduce its maintenance costs. When compared to an SPM strategy, while not visualized, we also find that the value of our model is inversely related to the machine usage intensity. This increased value of possibly advancing maintenance for lower usage intensities align with results that a

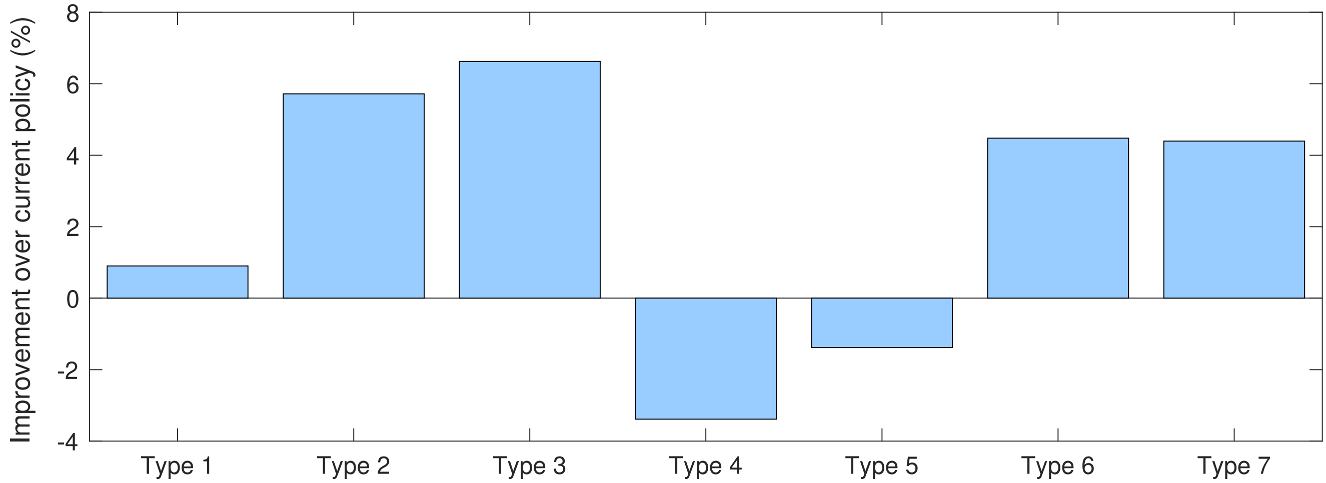


Fig. 4. Comparison of improvement in the expected cost per decision epoch between our policy and the current policy.

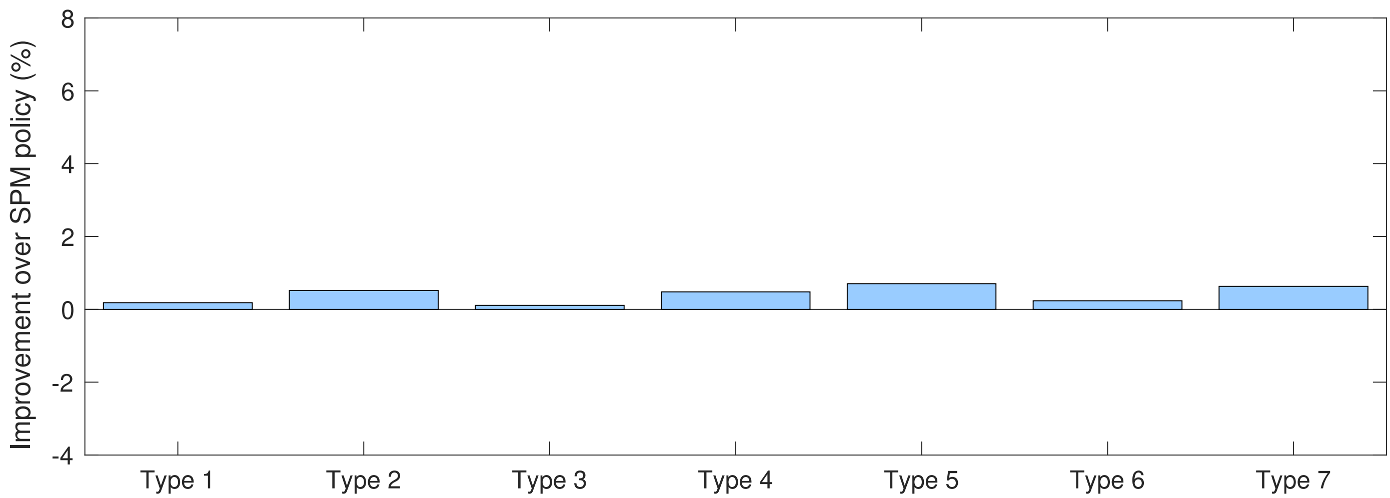
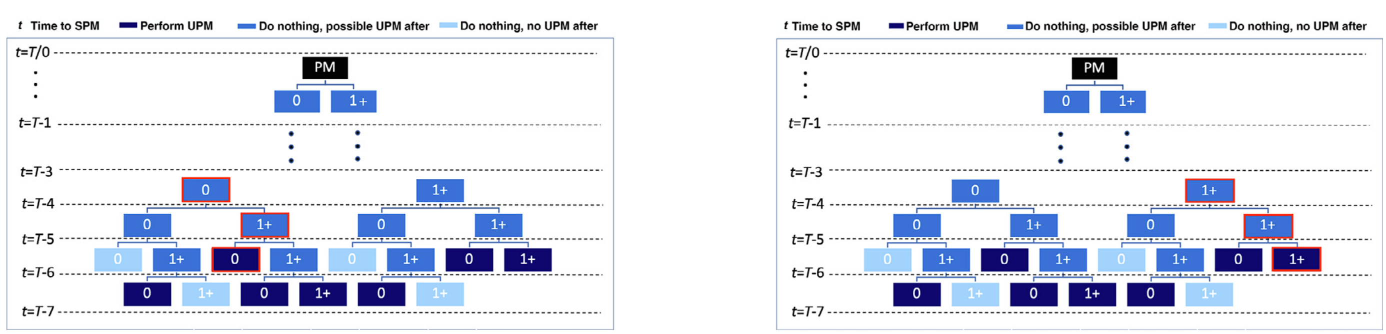


Fig. 5. Comparison of improvement in the expected cost per decision epoch between our policy and the SPM only policy.



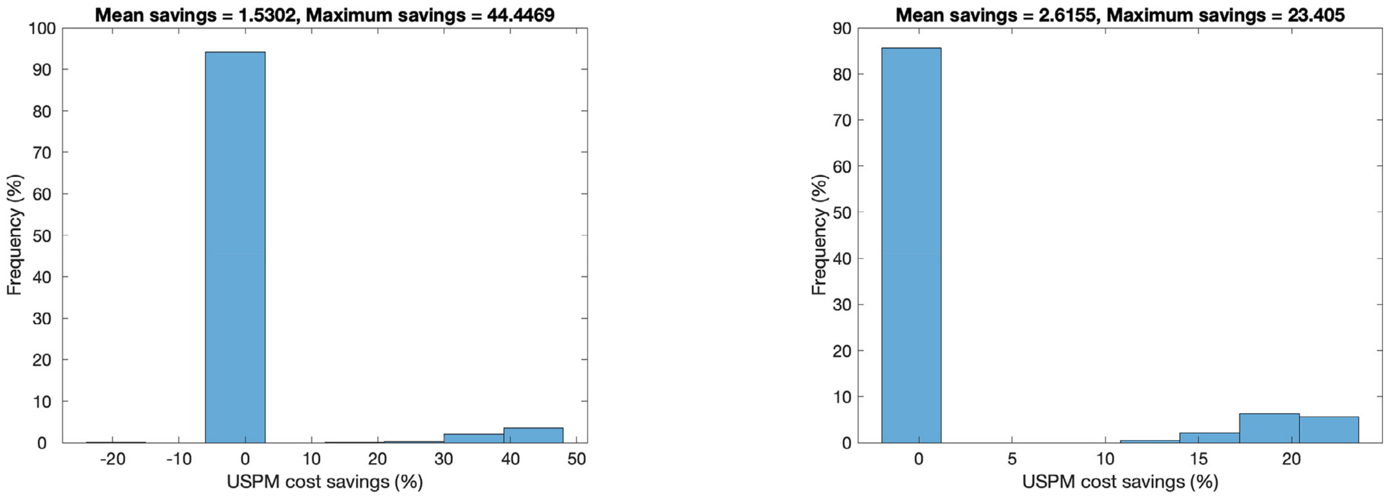
(a) Sample path 1.

(b) Sample path 2.

Fig. 6. We compare the expected costs per decision epoch of two different sample paths for a machine class USPM optimal policy (from Fig. 2) with the SPM policy.

low intensity machine's maintenance interval should generally be shorter (also supported by Fig. C.12).

We reiterate that the results reported in this section, when benchmarking our USPM policy against the SPM policy, aligns with the periodic maintenance interval provided by the OEM. It is possible that this interval is not the true optimal. Nevertheless, the current analysis shows the value of a dynamic maintenance policy (our USPM policy) over a scheduled periodic maintenance only policy (the SPM policy). We omit a dedicated analysis determining the optimal periodic maintenance interval, but we note that numerical exploratory analyses reveal that both the USPM and SPM policies perform worse than the current policy for shorter inter-



(a) Sample path 1.

(b) Sample path 2.

Fig. 7. Expected cost savings of our USPM policy over the SPM policy for the two sample paths of Fig. 6.

vals. For longer intervals, our USPM policy outperforms both the SPM and the current policies at an increasing rate. The results suggest that the interval used is likely close to the optimal one. However, the reliability of our results goes down as the periodic maintenance interval is extended, due to the small-data problem combined with the fact that we have less data available for longer periodic maintenance intervals. Regardless, due to its flexibility, our USPM policy will never perform worse than the SPM policy.

5. Conclusions

In this work, we investigate how failure and maintenance data may be used to obtain preventive maintenance prescriptions. We combine data pooling using a Poisson GLM with an adapted variable order Markov model to determine the most likely transitions between failure states for different machine classes and usage intensities in a periodic maintenance interval. The resulting probabilities are used as input to an MDP with the aim of minimizing the expected maintenance costs over a contract horizon. The policies obtained from the MDP prescribe when it is optimal to perform unscheduled preventive maintenance. We apply our model to a data set with failure and preventive maintenance logs of 3 000 machines. On average, our policies outperform those currently being implemented by 5% and a scheduled periodic maintenance only policy by 0.8% and up to 44%, for the machine class and sample paths considered. Our model may therefore be used to flag bad machines which exhibit irregular failure patterns. Our model's application, however, is limited to scenarios for which scheduled periodic maintenance may be rescheduled, as is the case for our partner OEM.

We train our model on failure and maintenance data and assume that the future will evolve similarly. While this assumption is a limitation of our work, our model is expected to flag the majority of irregular failure patterns early, allowing for proactive intervention. Additionally, our model can easily be retrained should new failure patterns emerge. Future work may include incremental learning of the most likely future failure patterns, given new data. Given availability of the required failure and maintenance data, our model may easily be generalized to different machines and machine profiles for which such data is available.

We follow a "predict, then optimize" approach in this paper. While our prescription model performs relatively well in terms of robustness for the performed analyses, Elmachtoub & Grigas (2020) showed that integrating the prediction model into the prescription model provides opportunities to further improve the resulting prescriptions. Future work may include extending our analysis to such an approach. Finally, future work may determine the effects on the optimal policy when a threshold determining whether future preventive maintenance interventions should be rescheduled or not, is imposed.

Appendix A. Notation list

We include a list of frequently used notation in Table A.1.

Appendix B. The data set

Our data set source is the same as that of Deprez et al. (2020) in which the objective is to price maintenance contracts based on the predicted periodic maintenance costs. In our paper, we now improve on the periodic maintenance schedule (to reduce

Table A.1
The list of frequently used notation abbreviations.

Symbol	Meaning	Symbol	Meaning
N	Maintenance contract length	$a \in \{SPM, UPM, NPM\}$	Action set
T	Decision epochs between SPMs	$s_n \in S$	Failure occurrences in epoch n
t	Decision epochs until next SPM	$\mathbf{s} = [s_{n-l}, \ldots, s_{n-1}]$	Failure vector
τ	Decision epoch length	L	Look-back period
$C_P(a)$	Preventive maintenance cost given a	l	$\min\{T-t, L\}$
$C_F(s)$	Corrective maintenance cost given s	$p_t^a(s\|\mathbf{s})$	Failure probability

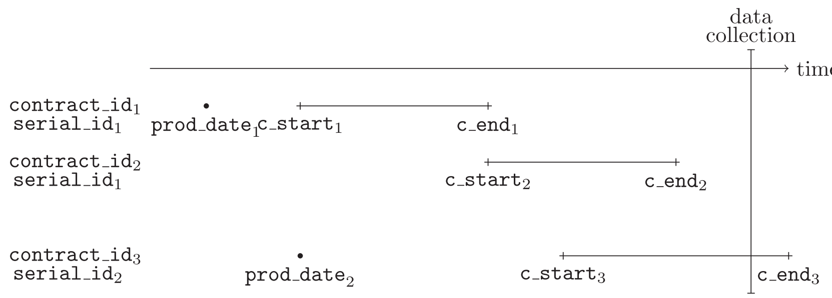


Fig. B.8. For each contract, the data set contains the contract number, the machine number, the machine's production date, and the start and end date of the contract. contract_id$_2$ is a contract renewal for machine serial_id$_1$. contract_id$_3$ is still active at the extraction time of the data (duplicated from Deprez et al. (2020) with permission).

the maintenance costs). We refer to the paper of Deprez et al. (2020) for a more in-depth discussion on the data and highlight here our main pre-processing steps.

B1. The maintenance contracts data set

Figure B.8 illustrates the structure of the available data over time. Table B.2 lists the available variables.

Contract information Each contract is labeled by a unique number, `contract_id`, characterized by a start and end date, i.e. `c_start` and `c_end` respectively, see Fig. B.8. The contracts in our data set start between 2004–2020 and about 66% of the contracts terminate within that window. The contract duration, Δ t = c_end-c_start, ranges between two months and 16 years. A duration of 5 years is most common.

Machine information Each contract covers a single machine identified by a serial number, `serial_id`. A contract can be a renewal on the same machine (see Fig. B.8), with a different contract number but the serial number remains unchanged. The data set comprises seven different machine classes, `m_class`. The production date, `prod_date`, is available for about 89% of machines. For the remaining machines, we use the installation date or, if unavailable, the initial contract start date, as a reasonable imputed production date. *Maintenance visits* Maintenance visits are logged for each contract. To disguise the frequency of maintenance interventions, we split the contract duration into a number of contract decision epochs. A decision epoch is a period of η running hours, depending on the MDP model specification. Each maintenance visit is characterized by a start- and end-of-service date, `v_start` and `v_end`. The former is used to determine the epoch in which the visit was executed. We keep track of the calendar year in which the visit was performed, `calendar_yr`. The cost of each maintenance visit and type are respectively logged in `v_cost` and `v_type`. The maintenance visits are categorized into scheduled preventive maintenance and corrective maintenance, denoted as PM and failure, respectively. The total running hours of a machine, since its initial deployment, are logged upon the maintenance visit as v_RHS. This initial deployment is not necessarily the start of the (current) contract. Fig. B.9 visualizes the timeline of one specific contract in our data set with a contract duration of 4.5 years. The different service visits are displayed over time with their respective costs and types, represented by the height and the color of the bars, respectively. Where available, the total running hours since the initial deployment or installation of the machine are registered with a black dot at a visit.

B2. Data pre-processing

We perform the following data pre-processing steps. *Running hours* The running hour v_RHS data is missing for about 50% of the maintenance interventions in our data set, see for instance Fig. B.9. We impute the missing values in two ways. We use linear interpolation if running hours were measured at both an earlier and later point in time, and linear regression to extrapolate the running hours beyond the last registered measurement. We base the interpolation and extrapolation on v_start, the start date of the visit. These approaches assume some linearity of running hours over time. We therefore discard all machines for which the linear regression has $R^2 < 0.85$. We also discard machines with two or less running hour measurements, as they do not allow for a decent assessment of the running hours. By implementing these selection rules, we removed about 25% of the contracts from the initial data set, resulting in about 5 000 different contracts.

We add the variable RHS_av based on the running hour data. RHS_av is the slope of the regression model and represents the average number of running hours per year and consequently the machine usage intensity.

Contract length and periodic maintenance frequency The data set contains contracts with significantly long periodic maintenance intervals. These contracts are, in our opinion, not representative of the OEMs maintenance policy and, as such, we remove these contracts from the data set. Specifically, we discard contracts which contain at least one periodic maintenance interval larger than the mean maintenance interval plus two standard deviations. Further, we observe very long or very short (in running hours) contracts in the data set. In our opinion, these specific contracts are not representative of the average contract and are consequently removed. We discard contracts with a running hour duration shorter than the mean duration minus one standard deviation or longer than the mean duration plus two standard deviations. The asymmetry in our selection rule has to do with the asymmetry of the distribution of contract lengths. Finally, we discard contracts whose length, in running hours, is less than two periodic maintenance intervals as set by the MDP.

Maintenance visit costs To cope with the long tail of the failure cost data, we omit the highest 0.1% of the failure costs. Moreover, in cooperation with the OEM, we cleaned the visit cost data as follows. First, we removed visit costs below a predetermined minimum cost. These costs are assumed to be wrong since there is a minimal setup cost for a maintenance visit. Second, some of the preventive maintenance cost values seemed unrealistically high and were deemed mislabeled or wrong. To resolve this issue, we removed the one percent highest values for preventive maintenance visits. We account for cost inflation using the strategy outlined in Deprez et al. (2020) and inspired by Lopez (2019). This leads to an inflation rate of $r = 1.94\%$ for the reference year of 2010. Using this inflation rate, all costs are adjusted to 2010 currency. We omit the actual costs determined for each machine due to confidentiality reasons.

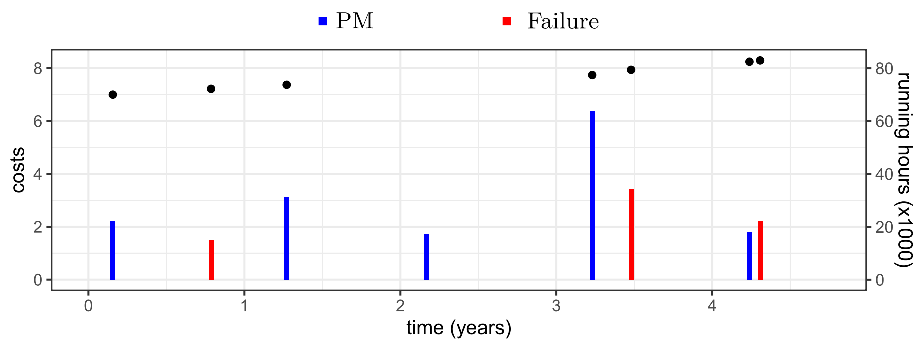


Fig. B.9. Example of a contract of 4.5 years with the time of interventions, costs and (if available) total running hours at the moment of intervention. All costs re-scaled for reasons of confidentiality. (adapted from Deprez et al. (2020)).

Table B.2
Definition of the variables available in the data set (adapted from Deprez et al. (2020)).

	Name	Description	Values
contract	`contract_id`	unique identification number of the contract	confidential
	`c_start`	start date of the contract	date between 2004–2020
	`c_end`	end date of the contract	date between 2006–2020
	Δ t	duration of the contract	[0,16]
machine	`serial_id`	unique identification number of the machine	confidential
	`m_class`	machine class	{Type 1, Type 2, . . . , Type 7}
	`prod_date`	estimated production date of the machine	date between 1976–2018
visit	`v_start`	start date of the visit	date between 2004–2020
	`v_end`	end date of the visit	date between 2004–2020
	`calendar_yr`	calendar year at the visit time	{2014, 2015, . . . , 2020}
	`epoch`	period of the contract the visit is executed	confidential
	`v_cost`	total costs of the visit	confidential
	`v_type`	visit type	{PM, Failure}
	`v_RHS`	total running hours at visit since initial deployment of the machine	

Table B.3
The frequency factors used for the 0/1+ and 0/1/2+ models. The frequency 1+ column represents the expected number of failures in an epoch where a failure occurred. The frequency 2+ column represents the expected number of failures in an epoch when at least two failures occurred.

Machine Class	Frequency 1+	Frequency 2+
Type 1	1.2559	2.2755
Type 2	1.1848	2.2576
Type 3	1.2524	2.3733
Type 4	1.1671	2.2607
Type 5	1.1987	2.2038
Type 6	1.1997	2.2457
Type 7	1.2076	2.2210

For the corrective maintenance costs, we include an additional downtime cost for unplanned lost productivity as a ratio of the corrective maintenance costs. To allow for comparison of results with current practices and across maintenance intervals, we multiply the corrective maintenance costs with a frequency factor. The frequency factors, as shown in Table B.3, are determined as the expected severity of failures for an epoch in which a failure occurs.

Table C.4
Machine usage intensity between and within machine classes are spread unevenly.

Machine Class	Low	Medium	High
Type 2	37.5%	37.2%	25.3%
Type 5	26.0%	32.0%	42.0%
Type 7	54.8%	31.9%	13.3%

Appendix C. Descriptive statistics

We include descriptive statistics which support our analysis, in Figs. C.10 to C.12. In the figures, RHS refers to machine running hours.

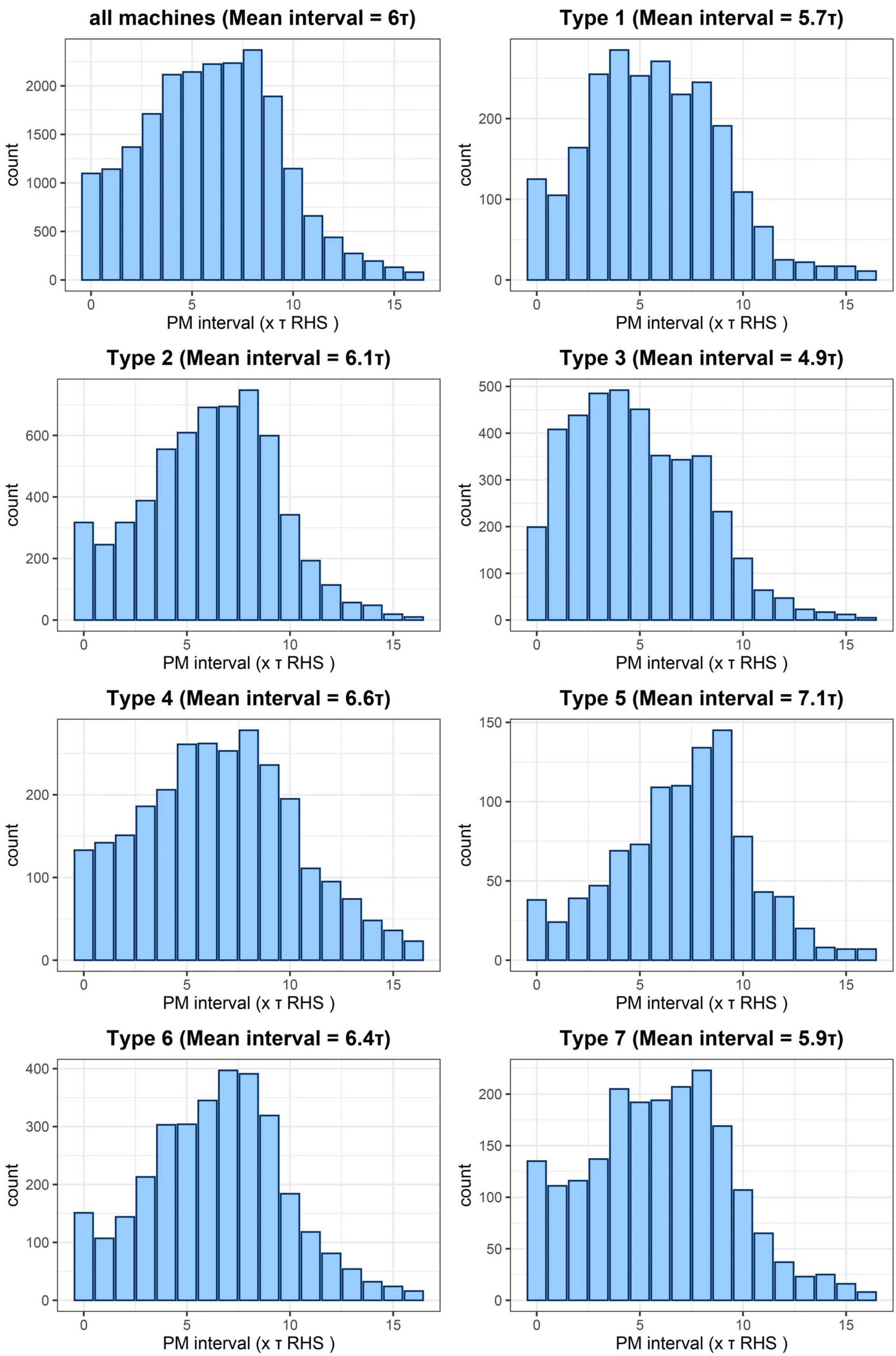


Fig. C.10. The frequency of periodic maintenance (PM) intervals for the various machine classes.

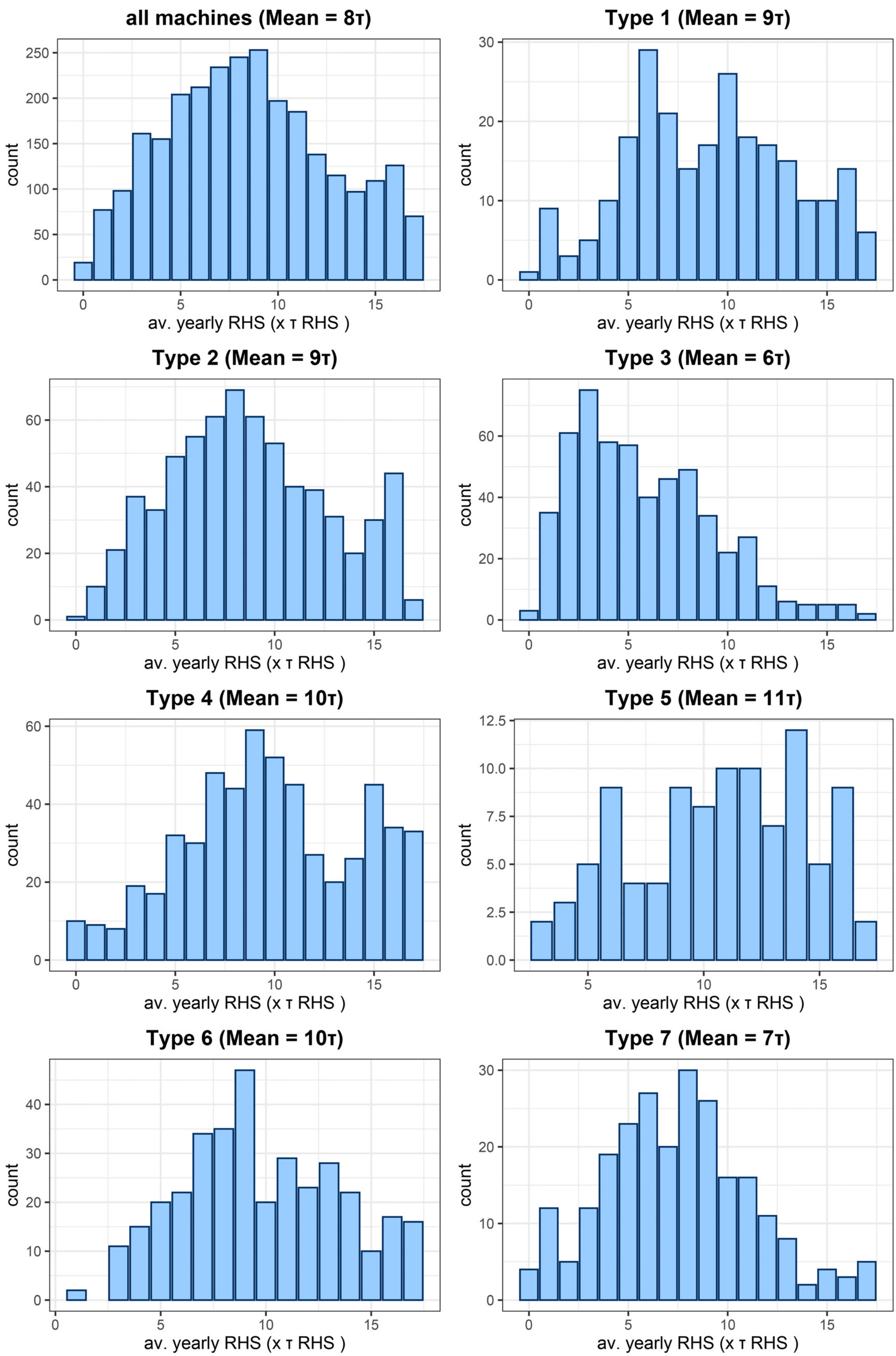


Fig. C.11. Average running hours (RHS) per machine, showing distribution of usage intensity between and within machine classes.

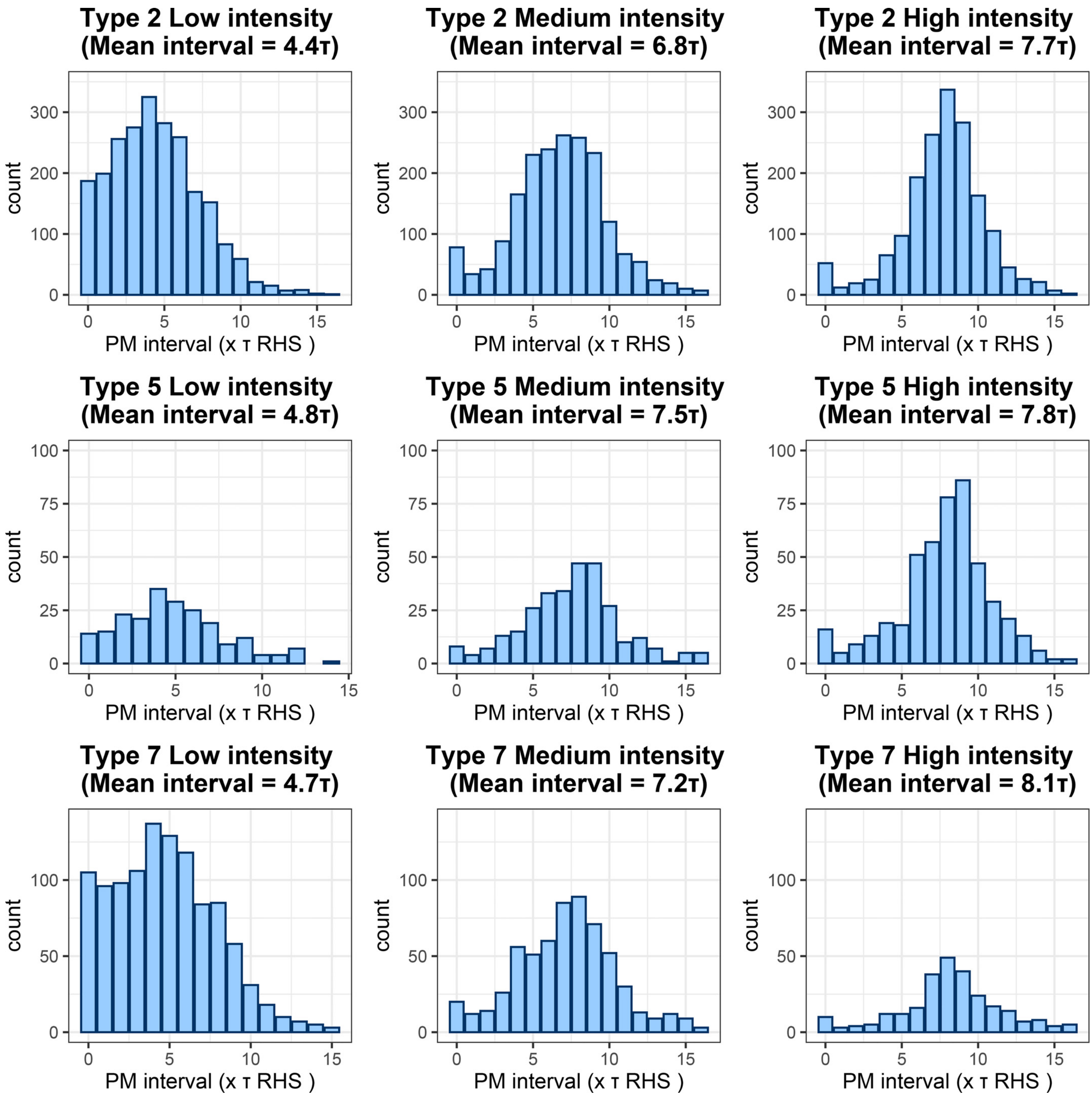


Fig. C.12. Periodic maintenance (PM) interval frequency for machine classes Type 2, 5 and 7 according to intensity of use.

Appendix D. Model parameter estimation

Decision epoch length and failure states per epoch Fig. D.13 shows, for our data set, the distribution of the failure observations in a decision epoch for three different decision epoch lengths, η, 2η and 4η.[1] The number of expected failures in a decision epoch decreases the shorter the decision epoch length, as shown in Fig. D.13. The flexibility in performing unscheduled preventive maintenance, enabled by a shorter decision epoch length, therefore allows the decision maker to be more proactive in avoiding unexpected failures. A shorter decision epoch length, however, in turn increases the number of decision epochs in a periodic maintenance interval and, possibly, the number of the look-back periods, L. Increasing L effectively increases our model's demand for data of failure combination instances. The latter reduces the accuracy of the transition probability matrices as only few such observable instances may be available.

Increased flexibility of a short decision epoch length therefore comes at a cost of potential decreased transition probability accuracy. In contrast, a longer decision epoch length results in both decreased flexibility in performing unscheduled preventive maintenance and an increase in the number of failure states to consider to ensure relevant prescriptions. For example, to consider only two

[1] For confidentiality reasons, the exact decision epoch length is disguised.

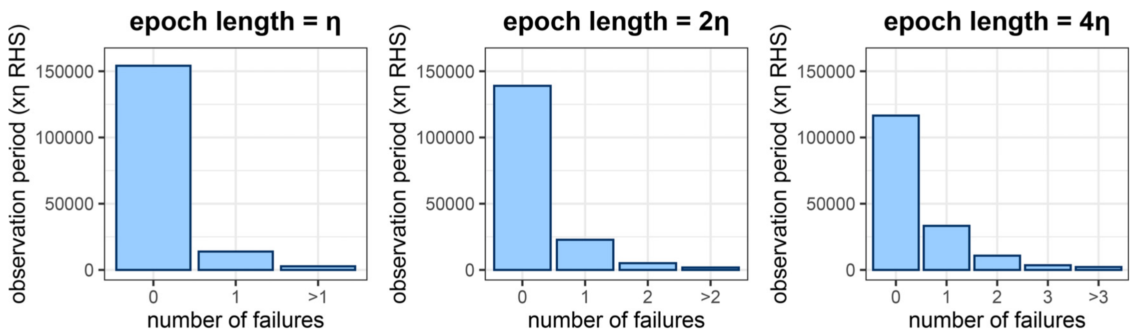


Fig. D.13. The frequency of failures per epoch length.

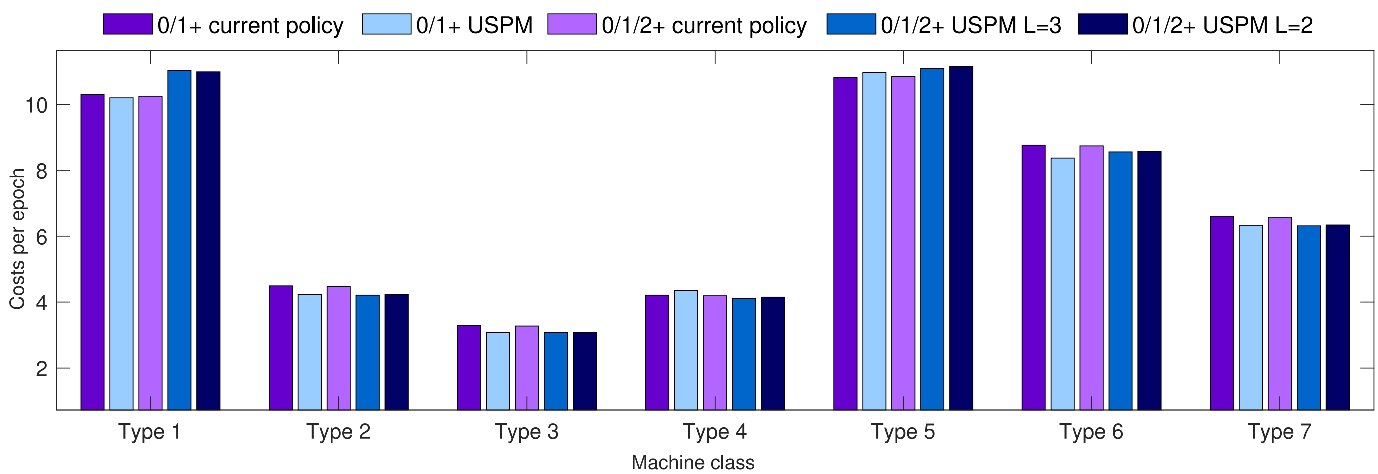


Fig. D.14. Comparison of the expected cost per decision epoch for the current policy and our model for failure states 0/1+ and 0/1/2+ for a periodic maintenance interval of eight decision epochs.

failure state combinations (0/1+) for a decision epoch length of Fig. D.13 (c), may result in sub-optimal maintenance prescriptions in comparison to a five failure state combination (0/1/2/3/4+). Naturally, for a constant L, an increase in failure states considered exponentially increases the state space of the MDP. Known as the curse of dimensionality, such an MDP is complex to solve and it may be difficult to obtain interpretable prescriptions from the results.

Considering the failure state and estimation accuracy trade-off due to the decision epoch length, we jointly decide on the decision epoch length and number of failure states to consider. We fix the epoch length at running hours option (a) (Fig. D.13), such that $\tau = \eta$, and two failure states per epoch, namely no failures (failure state 0) or at least one failure (failure state 1). The choice limits the average number of machine failures in an epoch at approximately 1 (Fig. D.13) and allows us to limit failure estimation accuracy loss.

We next investigate our model's robustness by performing a sensitivity analysis on the effect of our choice in the number of failure states considered on obtained maintenance costs. Considering more than two failure states in our model will allow for advancing maintenance based on different failure intensities per decision epoch. However, increasing the failure states simultaneously increases the model's demand for data, with an expected decrease in prediction accuracy. We therefore expect to see a noted difference in expected costs for our two failure state (0/1+ failures) policy when compared to the three failure state (0/1/2+ failures) policy. In order to investigate the cost differences (and perceived prediction accuracy), we consider two look-back values, $L = 2$ (9 resulting failure state combinations) and $L = 3$ (27 resulting failure state combinations). The results are graphed in Fig. D.14.

We use the frequency factors from Table B.3 to determine the costs for the three failure state (0/1/2+ failures) policy. By using the frequency factors of Table B.3, the average costs for the two failure state (0/1+ failures) current policy and that for the three failure state (0/1/2+ failures) current policy should be approximately equal, which is the case as can be seen in Fig. D.14. From Fig. D.14, we find that the policy considering more failure states (0/1/2+ failures) typically results in only minor differences in expected costs. We only record a significant increase in expected costs for the Type 1 machine class. The results suggest that the model may generally be robust to small changes in the number of failure states considered.

Periodic maintenance interval The periodic maintenance interval frequency per machine class is shown in Appendix C, Fig. C.10. From the figure, the mean maintenance interval length for the majority of the machine classes are between six and seven decision epochs. The maintenance interval length recommended by the OEM, however, is eight decision epochs. A current mean interval length shorter than that recommended by the OEM reveals that the OEM occasionally deviates from their fixed schedule of eight decision epochs. We note that the OEMs deviation may be driven by technician availability or performing unscheduled preventive maintenance as a result of suspicious machine behavior. We do not have data on the actual reasons.

In our model, we therefore set the maintenance interval for all machine classes to be a multiple of our decision epoch length closest to the overall mean historical interval length across all machine classes, which is six, plus two epochs. The result is an interval of eight decision epochs. The additional two epochs allow us to capture the effect of advancing scheduled maintenance while

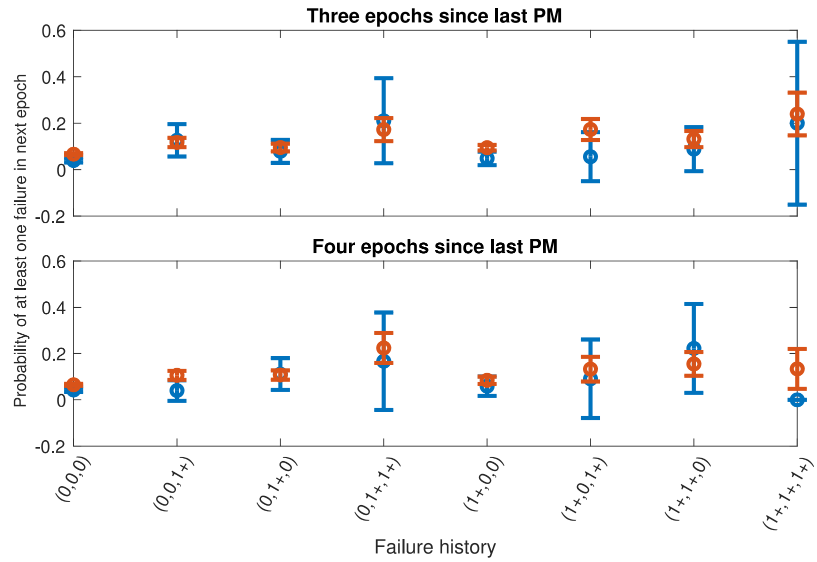


Fig. E.15. The failure probability point estimates and their respective margins of error for a machine, for respectively three and four decision epochs since the most recent preventive maintenance action. The predictions from the original data set are shown in blue and those from the transformed data set are shown in red. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

balancing the requirement for sufficient data. We do not allow for maintenance intervals longer than our predetermined one, which is consistent with pure periodic maintenance and expert recommendations for maintenance of critical machines.

Contract length To allow for comparison with current practices, we set the contract length, N, of all machine classes, in running hours, equal to the mean historical length of all the contracts for that machine class, as observed from the data set.

Appendix E. Transition probability matrix estimation using the failure prediction model

We populate the transition probability sets for each machine class and usage intensity combination using the data pooling approach of Section 3.2. We determine the maximum look-back period in a maintenance interval, L, as 3. This value is the highest look-back value for which the standard sample error over the failure states (which increases as L increases) first exceeds 5%. From the standard errors used to determine L, we obtain 95% confidence intervals (margins of error) for the different failure probability point estimates. We plot the results in Fig. E.15 for $L = 3$ for two different decision epochs in the periodic maintenance interval, namely three and four decision epochs since the most recent preventive maintenance action was performed ($t = T - 3$ and $t = T - 4$) for a machine class. The empirical estimates for the original data set (decoupled per machine class) is shown in blue, while those from the transformed data set (pooled across machine classes) are shown in red.

We find that the resulting 95% confidence intervals, based on the conventional normality assumption, for the transformed data sets are generally smaller than those for the original data sets. Smaller confidence intervals reflect less variation in the predictions. Exceptions are failure histories for which there was no record in the original data set, such as failure history $\mathbf{s} = [1+, 1+, 1+]$ and a resulting zero probability of failure for four decision epochs since the last preventive maintenance action ($t = T - 4$). Additionally, for some probabilities determined by means of the original data set, such as failure history $\mathbf{s} = [0, 1+, 1+]$ four decision epochs since the last preventive maintenance action ($t = T - 4$), the lower limit of the 95% confidence interval is negative. We attribute the negative probability, which is not possible, to the normality assumption.

From Fig. E.15, we additionally see that the confidence intervals of the transformed data set failure probabilities generally fall within those of the original data set. The results further support the use of the transformed data to predict the probability of future machine failures. Where the confidence intervals of the transformed data set do not fall within the confidence intervals of the original data set, we note that it is the upper limit from the original data set that is exceeded. The results suggest that the probability of a machine failure may be underestimated when using the original data set only. The maintenance policies that result from using the transformed data is therefore expected to be more robust. Nonetheless, since we determine the weights using the GLM point estimates, we next investigate the effect on our model's robustness when instead drawing these weights from the estimated Poisson GLM distribution.

Specifically, we obtain 100 different weights by sampling 100 estimates of the Poisson GLM, which estimates the Poisson parameter, β_x, for each machine class x (Section 3.2), as follows. We obtain the probabilities $P(s|x)$ and weights $w(s)_{y,x}$ (see Eq. (6)) using the Poisson GLM point estimates. Using the point estimates and the standard errors of the point estimates as the mean and standard deviation of a normal distribution, respectively, we sample 100 β_x's for each machine class x. Each sample of β_x's enables us to calculate a set of weights used to pool the data and ultimately determine the associated transition matrices. The process results in 100 samples of the transition probability matrices used in our MDP. We illustrate the resulting policy costs, relative to the current policy, for Type 2 and 7 machine classes (disregarding usage intensity for visualization purposes) and the 100 different transition matrices in Fig. E.16. The improvement of the base MDP policy over the current policy is shown in green. From the figure, our policy is expected to improve on the current policy by at least 3% and 5% for the Type 2 and Type 7 machine classes, respectively.

From Fig. E.16, the spread in expected costs, relative to the current policy, is constant at approximately 3.5% for the Type 7 machine and at approximately 2% for the Type 2. The Type 7 machine data accounts for less than 10% of our data set, while the Type 2 accounts for more than 25%. Consequently, we expect that a change in weights will have a larger effect on the optimal policy

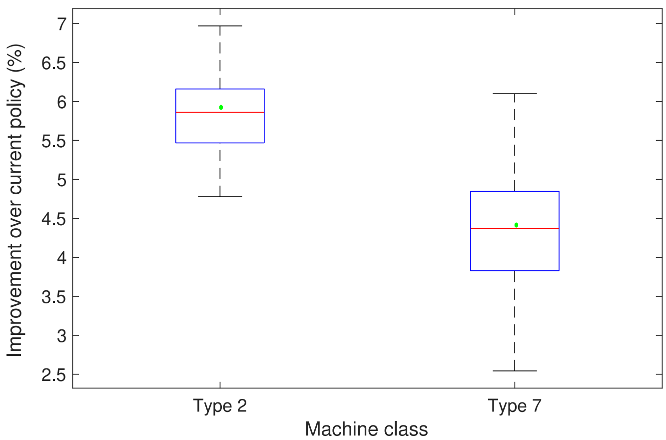


Fig. E.16. We sample 100 weights from the GLM Poisson model and compare the resulting policy costs with that of the current policy for two machines. The model performs consistently well, with the majority of deviations within 1% from the base case for both machines. For each case, the base policy cost deviation is shown in green. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

and related costs for a Type 7 machine (with a small original share of the complete data set) than for a Type 2 (with a large original share of the complete data set). Nonetheless, the relatively small spread in costs for the Type 2 machine suggests that the weights used to pool the data provide reliable transition probability estimates. Additionally, for each of the two machine classes considered, we do not observe a significant change in the prescribed decisions by the obtained optimal policies when compared to the point estimate policy. In most cases, the optimal policy will deviate with one or two additional unscheduled maintenance actions. Combined with the results in Fig. E.16, the small changes in the obtained policies suggest that the optimal point estimate policy is fairly robust to uncertainty in the weights used to estimate the transition probability matrices.

References

Barlow, R. E., & Hunter, L. C. (1960). Optimum preventive maintenance policies. *Operations Research, 8*, 90–100.
Bastani, H., Zhang, J. D., & Zhang, H. (2020). Applied machine learning in operations management. In V. Babich, J. Birge, & G. Hilary (Eds.), *Innovative technology at the interface of finance and operations*. In *Springer series in Supply Chain Management*. Springer Nature.
Boute, R. N., & Van Mieghem, J. A. (2021). Digital operations: Autonomous automation and the smart execution of work. *Management and Business Review, 1*(1), 177–186.
Buhlmann, P., & Wyner, A. J. (1999). Variable length Markov chains. *Annals of Statistics, 27*(2), 480–513.
Coleman, C., Damodaran, S., Chandramouli, M., & Deuel, E. (2017). Making maintenance smarter: Predictive maintenance and the digital supply network. *Technical Report*. Deloitte.
De Jonge, B., & Scarf, P. A. (2020). A review on maintenance optimization. *European Journal of Operational Research, 285*, 805–824.
Deprez, L., Antonio, K., & Boute, R. N. (2020). Empirical risk assessment of maintenance costs under full-service contracts. *Working Paper*. https://doi.org/10.13140/RG.2.2.19444.68488.
Drent, C., Kapodistria, S., & Boxma, O. (2020). Censored lifetime learning: Optimal Bayesian age-replacement policies. *Operations Research Letters, 48*, 827–834.
Drent, C., Kapodistria, S., & Resing, J. (2019). Condition-based maintenance policies under imperfect maintenance at scheduled and unscheduled opportunities. *Queueing Systems, 93*, 269–308.
Dursun, I., Akay, A., & Van Houtum, G. J. J. A. N. (2020). Age-based maintenance under population heterogeneity: Optimal exploration and exploitation. European Journal of Operational ResearchForthcoming.
Elmachtoub, A. N., & Grigas, P. (2020). Smart "predict, then optimize". *Working Paper*.
Gupta, V., & Kallus, N. (2020). Data-pooling in stochastic optimization. *Working Paper*.
Haarman, M., de Klerk, P., Decaigny, P., Mulders, M., Vassiliadis, C., & Sijtsema, H. (2018). Predictive maintenance 4.0. beyond the hype: PdM 4.0 delivers results. *Technical Report*. PwC and Mainnovation.
Ivanov, D., Tang, C. S., Dolgui, A., Battini, D., & Das, A. (2020). Researchers' perspectives on Industry 4.0: Multi-disciplinary analysis and opportunities for operations management. International Journal of Production Research. Forthcoming.
Kim, M. J., & Makis, V. (2013). Joint optimization of sampling and control of partially observable failing systems. *Operations Research, 61*(3), 777–790.
Kumwilaisak, W., & Kuo, C. J. (2002). Adaptive variable length Markov chain for non-stationary fading channel modeling. In *Globecom conference, ieee: vol. 3* (pp. 2046–2050).
Lopez, O. (2019). A censored copula model for micro-level claim reserving. *Insurance, Mathematics & economics, 87*, 1–14.
Lueth, K. L., Patsioura, C., Williams, Z. D., & Kermani, Z. Z. (2016). Industrial analytics 2016/2017: The current state of data analytics usage in industrial companies. *Technical Report*. Digital Analytics Association Germany and IoT Analytics.
Manyika, J., Chui, M., Bisson, P., Woetzel, J., Dobbs, R., Bughin, J., & Aharon, D. (2015). The internet of things: Mapping the value beyond the hype. *Technical Report*. McKinsey Global Institute.
Mišić, V. V., & Perakis, G. (2019). Data analytics in operations management: A review. *Manufacturing & Service Operations Management, 22*(1), 158–169.
Ohlsson, E., & Johansson, B. (2010). *Non-life insurance pricing with generalized linear models*: vol. 2. Springer.
Poppe, J., Boute, R. N., & Lambrecht, M. R. (2018). A hybrid condition-based maintenance policy for continuously monitored components with two degradation thresholds. *European Journal of Operational Research, 268*(216), 515–532.
Puterman, M. L. (1994). *Markov decision processes: Discrete stochastic dynamic programming*. Wiley, NY.
Regattieri, A., Manzini, R., & Battini, D. (2010). Estimating reliability characteristics in the presence of censored data: A case study in a light commercial vehicle manufacturing system. *Reliability Engineering & System Safety, 95*(10), 1093–1102.
Sanoubar, S., He, K., Maillart, L. M., & Prokopyev, O. A. (2020). Optimal age-replacement in anticipation of time-dependent, unpunctual policy implementation. *IEEE Transactions Reliability*, 1–16.
Sgarbossa, F., Zennaro, I., Florian, E., & Calzavara, M. (2020). Age replacement policy in the case of no data: The effect of Weibull parameter estimation. *International Journal of Production Research, 58*(19), 5851–5869.
Van Oosterom, C. D., Elwany, A. H., Çelebi, D., & Van Houtum, G. J. (2014). Optimal policies for a delay time model with postponed replacement. *European Journal of Operational Research, 232*(1), 186–197.
Van Staden, H. E., & Boute, R. N. (2021). The effect of multi-sensor data on condition-based maintenance policies. *European Journal of Operational Research, 290*(2), 585–600.
Zhu, Q., Peng, H., Timmermans, B., & van Houtum, G. J. J. A. N. (2017). A condition-based maintenance model for a single component in a system with scheduled and unscheduled downs. *International Journal of Production Economics, 193*, 365–380.