

Condition-Based Maintenance for Complex Systems based on Current Component Status and Bayesian Updating of Component Reliability

Gero Walter^a, Simme Douwe Flapper^a

^a*School of Industrial Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands*

Abstract

We propose a new condition-based maintenance policy for complex systems, based on the status (working, defective) of all components within a system, as well as the reliability block diagram of the system. By means of the survival signature, a generalization of the system signature allowing for multiple component types, we obtain a predictive distribution for the system survival time, also known as residual life distribution, based on which of the system's components currently function or not, and the current age of the functioning components.

The time to failure of the components of the system is modeled by a Weibull distribution with a fixed shape parameter. The scale parameter is iteratively updated in a Bayesian fashion using the current (censored and non-censored) component lifetimes. Each component type has a separate Weibull model that may also include test data.

The cost-optimal moment of replacement for the system is obtained by minimizing the expected cost rate per unit of time. The unit cost rate is recalculated when components fail or at the end of every (very short) fixed inter-evaluation interval, leading to a dynamic maintenance policy, since the ageing of components and possible failures will change the cost-optimal moment of replacement in the course of time. Via numerical experiments, some insight into the performance of the policy is given.

Keywords: condition-based maintenance, system reliability, remaining useful life, survival signature, unit time cost rate

1. Introduction

Both in practice and academics, there is a growing interest in condition-based maintenance (CBM), see e.g., Olde Keizer et al. (2017) The central idea behind CBM is to maintain systems or components at exactly the right time,

Email addresses: g.m.walter@tue.nl (Gero Walter), s.d.p.flapper@tue.nl (Simme Douwe Flapper)

i.e., just before they fail, in order to keep their reliability high and operating costs low, using information about the actual condition of the systems or components. In this context a trade-off is made between the risk of failure during operation (which can lead to costly downtime: idle workforce, missed production, penalties, loss of reputation) and the costs of premature maintenance (wasting potential component or system lifetime, downtime cost, cost of executing unnecessary maintenance activities).

Among the reasons for the above interest are the importance of increasingly short and reliable delivery times, and decreasing profit margins due to worldwide competition. All kinds of technical improvements have made it possible to estimate the condition of systems as a whole as well as their underlying components, whereas the cost for these technologies is decreasing rapidly, making the use of CBM policies even more feasible.

Most CBM policies are based on a directly observable, continuously measurable condition or degradation signal, e.g., the amount of vibration in case of rotating equipment. Alternatively, such a signal or health status is constructed using indirect measurements to determine the remaining useful life or time to failure of a system, see, e.g., Ahmadzadeh and Lundberg (2014); Si et al. (2011).

In this paper, we propose a new kind of CBM policy for when no such *usual* degradation signal for the system is available, but where the status (working or not working) of the system's components can be monitored (quasi-) continuously. In this situation, one can use the system's reliability block diagram and information with respect to the status of its components to directly calculate the system residual life distribution (RLD), and base the maintenance policy for the system on this distribution. In fact, our CBM policy can also be seen as based on a multivariate degradation signal, where each component sends a binary signal, and the reliability block diagram is used for sensor fusion.

To calculate the system RLD, we use the survival signature, which allows to deal with different types of components each having an individual failure time model (see Section 3). Here we assume that components of different types fail independently, and that failure times of components of the same type are independent given the model parameters.

In our new policy, there are three types of triggers to review the system: (1) at the end of each very short time interval δ (where δ is very short compared to the lifetimes of components), (2) directly after the failure of a component where the system still functions, and (3) directly after the failure of a component due to which the system as a whole no longer can function. We denote each review of the system in reaction to triggers (1) and (2) by 'evaluation', since the system RLD is re-evaluated at each of these time points, and a decision is made on whether preventive maintenance should be carried out or not. We use the term 'evaluation' in place of the term 'inspection', because the re-evaluation of the system RLD does not require a physical inspection, as we assume to learn of the exact failure times of components online via (quasi-) continuous monitoring.

In case of a trigger of type (1), the currently used models for component failure times are updated to account for the extra lifetime observed for the components. This information is taken into account via right-censored observations

in a Bayesian parameter update step. In case of a trigger of type (2), where a component has failed but the system is still functioning, in addition to the component models also the reliability block diagram of the system is updated. Based on the above changes, it is determined in both cases whether the system should be replaced now, or that this decision should be postponed until the next planned evaluation moment, accepting the risk of a system breakdown before. To determine the latter, we calculate the ‘optimal’ next evaluation moment based on minimizing the expected maintenance-related cost per unit of time for the present operational cycle. An operational cycle spans the time between two consecutive system maintenance actions. We assume that in both preventive and corrective maintenance, the entire system is replaced or brought to an as good as new state, and thus optimize the unit cost rate calculated over the time since the last system replacement.

When the ‘optimal’ next evaluation moment is before the next planned evaluation moment, we replace the system now; else we wait for the next planned evaluation moment. The reason behind this decision is that δ is taken as the shortest inter-evaluation period that can be realized from a hardware point of view. To illustrate, for components with lifetimes amounting to a few days, δ could be in the order of minutes; for components with lifetimes measuring in months, δ could be one day.

Note that strictly speaking, this decision procedure may not be optimal. However, because δ is very small when compared to the prior estimates of the lifetime of the components, at the time that the next ‘optimal’ evaluation moment is before the next planned evaluation moment, we usually already have done much better than without updating as described in this paper.

In this context it is also important to realize that our method doesn’t necessarily result in the actual optimal replacement solution for another reason, inherent to using updated information for decisions, when these updates can indicate that the former estimates were too optimistic or too pessimistic. The reason for this is that for some components, a too pessimistic prior probability density function (pdf) may lead to an increase in system reliability over time without any intervention. Whether or not this will be the case, as well as until when this kind of increase can occur, is uncertain at every moment in time. As a sidenote, this kind of uncertainty about the probabilistic behaviour itself is very often neglected in operations research literature. Determining the next ‘optimal’ evaluation moment can thus only indicate whether or not waiting with the replacement may result in lower or higher cost per unit of time, i.e., only indicates a local optimum, but not necessarily a global one.

In case of a trigger of type (3), where the failure of a component leads to the whole system failing, the currently used component failure time distributions are updated like in case of a trigger of type (1) and (2) to account for the information gained on components until the moment of system failure. Hereafter, a new system is installed, using the updated models for all components based on the insight obtained from the finished operational cycles so far.

At first sight, the above quasi-continuous updating of the component failure models, and the system reliability block diagram, as well as redetermining the

best moment to replace the system, seems to result in a (very) nervous maintenance policy. However, due to the very short calculation times, only at certain moments in time people involved in the execution of maintenance tasks need to receive detailed information, whereas during the remaining time, all can be handled automatically, as currently done within, e.g., many safety systems.

The setup of the rest of the paper is as follows. In Section 2, we describe our contribution to the CBM literature. Next, in Sections 3, 4 and 5, the three main steps in our policy are discussed in detail. Hereafter, in Sections 6 and 7, our policy is applied further to an example. Finally, in Section 8, a short summary and our main conclusions are given, followed by some suggestions for further research.

2. Literature review

Although there is a lot of literature on CBM policies for multi-component systems, see, e.g., Olde Keizer et al. (2017), we found only a few papers having some relation to our approach.

The paper that comes closest to this paper is the paper by Si et al. (2013). In that paper, the condition of a system is inspected, where the moment of the next inspection is determined via an updated degradation path based on the present condition of the system. The authors provide exact expressions for the RLD, also known as RUL (remaining useful lifetime) distribution, which is updated in an empirical Bayesian framework using conjugate priors. The RLD is used to construct a replacement decision model using a cost rate.

While we also use a Bayesian approach with conjugate priors to provide exact expressions for the RLD, one of the main differences between Si et al. (2013) and our paper is that we use component status data and the system reliability block diagram instead of a continuous degradation signal as basis for calculating the system RLD. Another main difference is that our policy can be applied to arbitrary system configurations (i.e., arbitrary reliability block diagrams with arbitrary components), whereas the method given by Si et al. (2013) can only be applied to single components. A third difference concerns how the cost rate is calculated. Si et al. (2013) calculate this rate using the renewal-reward theory approximation. Essential for using the renewal-reward theory for calculating the expected cost per unit of time is that the relevant characteristics of the system considered and its environment are not changing, i.e. that the failure behavior of the system, in essence its reliability function, does not change over time, and that the same policy with the same parameter values is used for a statistically long enough period of time. However, due to the update method related to the CBM policy presented in this paper, we cannot use the renewal-reward theory based calculation method, and therefore use the exact formula for the average cost per unit of time.

Another paper that includes some of the aspects covered in our paper is the paper by Sun et al. (2012). The authors construct a health index for a system based on sensor measurements, where the health status prediction is updated sequentially, resulting in a RUL distribution like with our method. However,

in contrast to our paper, the authors do not link their RUL distribution to any maintenance decisions.

Kim et al. (2011) developed a periodic monitoring CBM policy where a maintenance decision is triggered when a Bayesian control chart (a sequentially updated health indicator) exceeds a control limit (threshold) that is determined by minimizing the expected average cost per time unit. We use the same cost criterion, but base the maintenance policy decision directly on our exact RLD.

To summarize the main contribution of our paper, it introduces a CBM-like policy that does not require knowledge of the degradation path of the system, but can take into account continuous changes in the system reliability block diagram, as well as changes in the component failure time distributions.

3. Reliability function for complex systems using the survival signature

Let T_{sys} be the time to failure of the system, and $R_{\text{sys}}(t) := P(T_{\text{sys}} > t)$ be the system reliability function. Here we describe how $R_{\text{sys}}(t)$, given the system reliability block diagram and arbitrary component models, can be efficiently calculated using the survival signature.

We can analyze systems with an arbitrary reliability block diagram, consisting of components of K different types, where there are N_k exchangeable components of type k in the system, and N , the total number of components in the system, is given by $\sum_{k=1}^K N_k = N$.

As a running example, we consider a simplified automotive braking system consisting of four component types M , H , C and P , with the reliability block diagram as depicted in Figure 1. The master brake cylinder (M) activates all four wheel brake cylinders ($C_1 - C_4$), which in turn actuate a braking pad assembly each ($P_1 - P_4$). The hand brake mechanism (H) directly actuates the brake pad assemblies P_3 and P_4 , and the vehicle brakes when at least one brake pad assembly is actuated. Note that because of the ‘handbrake shortcut’, this system cannot be described as a nesting of series and parallel subsystems. Our example system is deliberately kept quite simple for didactical reasons. The survival signature allows to analyze much larger systems, see, e.g., the examples in Reed (2017), and our approach is able to deal with such systems.

In a system with N components, the state of the system can be expressed by the state vector $\mathbf{x} = (x_1, x_2, \dots, x_N) \in \{0, 1\}^N$, with $x_i = 1$ if the i th component functions and $x_i = 0$ if not. The structure function $\phi : \{0, 1\}^N \rightarrow \{0, 1\}$, defined for all possible \mathbf{x} , takes the value 1 if the system functions and 0 if the system does not function for state vector \mathbf{x} (Barlow and Proschan, 1975). Most real-life systems are coherent, which means that $\phi(\mathbf{x})$ is non-decreasing in any of the components of \mathbf{x} , so system functioning cannot be improved by worse performance of one or more of its components. Furthermore, one can usually assume that $\phi(0, \dots, 0) = 0$ and $\phi(1, \dots, 1) = 1$.

The survival signature (Coolen and Coolen-Maturi, 2012) is a summary of the structure function for systems with K groups of exchangeable components.

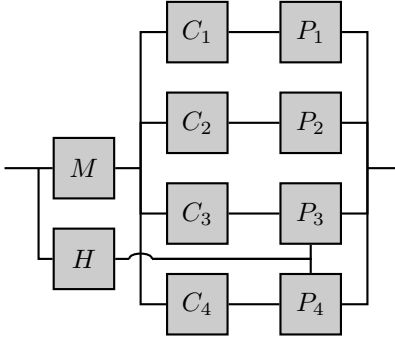


Figure 1: Reliability block diagram for a simplified automotive brake system with four component types M , H , C and P .

Denoted by $\Phi(l_1, \dots, l_K)$, with $l_k = 0, 1, \dots, N_k$ for $k = 1, \dots, K$, it is defined as the probability for the event that the system functions given that precisely l_k of its N_k components of type k function, for each $k \in \{1, \dots, K\}$. Essentially, this creates a K -dimensional partition for the event $T_{\text{sys}} > t$, such that $R_{\text{sys}}(t) = P(T_{\text{sys}} > t)$ can be calculated using the law of total probability,

$$\begin{aligned}
P(T_{\text{sys}} > t) &= \sum_{l_1=0}^{N_1} \cdots \sum_{l_K=0}^{N_K} P(T_{\text{sys}} > t \mid C_t^1 = l_1, \dots, C_t^K = l_K) P\left(\bigcap_{k=1}^K \{C_t^k = l_k\}\right) \\
&= \sum_{l_1=0}^{N_1} \cdots \sum_{l_K=0}^{N_K} \Phi(l_1, \dots, l_K) P\left(\bigcap_{k=1}^K \{C_t^k = l_k\}\right) \\
&= \sum_{l_1=0}^{N_1} \cdots \sum_{l_K=0}^{N_K} \Phi(l_1, \dots, l_K) \prod_{k=1}^K P(C_t^k = l_k), \tag{1}
\end{aligned}$$

where C_t^k is the number of type k components functioning at time t , and $P(C_t^k = l_k)$ is the (predictive) probability that exactly l_k components of type k function at time t . The last equality holds as we assume that components of different types fail independently. Note that for coherent systems, the survival signature $\Phi(l_1, \dots, l_K)$ is non-decreasing in each l_k .

Continuing our example, the survival signature for the system in Figure 1 is given in Table 1, omitting the entries for which $\Phi(l_M, l_H, l_C, l_P) = 0$ or $\Phi(l_M, l_H, l_C, l_P) = 1$, since the full table would contain $\prod_{k=1}^K (N_k + 1) = 2 \times 2 \times 5 \times 5 = 100$ rows. To illustrate how the table entries are obtained, consider the case $l_M = 0, l_H = 1, l_C = 0, l_P = 1$, i.e., M has failed, H functions, $C_1 - C_4$ have failed, and one of $P_1 - P_4$ functions. The system functions only if the single functioning type P component is either P_3 or P_4 . As the type P components are exchangeable, each of $P_1 - P_4$ has the same probability of being the functioning component, and so the probability of P_3 or P_4 being the functioning component is $2/4$, thus $\Phi(0, 1, 0, 1) = 0.5$. The omitted rows from Table 1 can be determined as follows:

l_M	l_H	l_C	l_P	Φ	l_M	l_H	l_C	l_P	Φ
1	0	1	1	0.25	1	0	2	1	0.50
1	0	1	2	0.50	1	0	2	2	0.83
1	0	1	3	0.75	1	0	3	1	0.75
0	1	0	1	0.50	1	1	0	1	0.50
0	1	0	2	0.83	1	1	0	2	0.83
0	1	1	1	0.50	1	1	1	1	0.62
0	1	1	2	0.83	1	1	1	2	0.92
0	1	2	1	0.50	1	1	2	1	0.75
0	1	2	2	0.83	1	1	2	2	0.97
0	1	3	1	0.50	1	1	3	1	0.88
0	1	3	2	0.83					
0	1	4	1	0.50					
0	1	4	2	0.83					

Table 1: Survival signature $\Phi(l_M, l_H, l_C, l_P)$ for the simplified automotive brake system depicted in Figure 1, omitting the rows for which $\Phi(l_M, l_H, l_C, l_P) = 0$ or $\Phi(l_M, l_H, l_C, l_P) = 1$.

- Take a row, decrease l_k for one $k \in \{M, H, C, P\}$ by 1. If the resulting row is not in Table 1, then the corresponding value of Φ is 0.
- Take a row, increase l_k for one $k \in \{M, H, C, P\}$ by 1. If the resulting row is not in Table 1, then the corresponding value of Φ is 1.

The R package `ReliabilityTheory` (Aslett, 2016) provides a convenient function to calculate the survival signature table based on any given graph.

4. Adaptive system residual life distribution based on Weibull component models

In this section, we describe how the system reliability function $R_{\text{sys}}^{(t_{\text{now}})}(t)$ at current time t_{now} can be calculated for a specific component model.

We consider the well-known Weibull model for the component lifetimes, which is used in a wide variety of reliability studies. To keep things simple, we assume that the shape parameter of the Weibull distribution is known, and that only the scale parameter needs to be estimated. The model could be extended to learn also the shape parameter in a later step, using, e.g., the discretized approach by Soland (1969).

By using the Bayesian approach described in Section 4.1, our component model allows to include both expert assessments and test data (if available), and furthermore can account for component lifetime information from previous operational cycles and the current operational cycle up to t_{now} . In Section 4.2 we derive the posterior predictive distribution for the Weibull component model as needed for the current system reliability calculation. Finally, in Section 4.3,

we adapt the method from Section 3 to the dynamic setting, resulting in a formula for $R_{\text{sys}}^{(t_{\text{now}})}(t)$.

The model description below follows closely (Walter and Coolen, 2017, §2, §4.1 – §4.3), who presented the same residual life distribution model. However, Walter and Coolen (2017) do not consider maintenance policies, but focus instead on a generalization of the RLD model by using imprecise probability models for the component lifetimes.

4.1. Bayesian update of the Weibull component models

Here we describe the Weibull model for the component lifetimes, together with the Bayesian update procedure which allows to include expert knowledge, component tests, and information from the monitored components in the system.

For each type k component, we assume that the lifetime $T_{k,i}$ ($i = 1, \dots, N_k$, $k = 1, \dots, K$) is Weibull distributed with scale parameter λ_k and (fixed) shape parameter $\beta_k > 0$, in short $T_{k,i} | \lambda_k \sim \text{Wei}(\beta_k, \lambda_k)$, with pdf (probability density function) and cdf (cumulative distribution function)

$$f(t_{k,i} | \lambda_k) = \frac{\beta_k}{\lambda_k} (t_{k,i})^{\beta_k - 1} e^{-\frac{(t_{k,i})^{\beta_k}}{\lambda_k}}, \quad (2)$$

$$F(t_{k,i} | \lambda_k) = 1 - e^{-\frac{(t_{k,i})^{\beta_k}}{\lambda_k}} = P(T_{k,i} \leq t_{k,i} | \lambda_k), \quad (3)$$

where $\lambda_k > 0$ and $t > 0$.

The shape parameter β_k determines whether the hazard rate is increasing ($\beta_k > 1$) or decreasing ($\beta_k < 1$) over time. For $\beta_k = 1$, one obtains the Exponential distribution with constant hazard rate as a special case. The scale parameter λ_k can be interpreted through the relation

$$E[T_{k,i} | \lambda_k] = \lambda_k^{1/\beta_k} \Gamma(1 + 1/\beta_k). \quad (4)$$

We will use this equation to convert expected lifetimes to λ_k and vice versa.

With a Bayesian approach, one can express expert knowledge about the reliability of the components by assigning a so-called prior distribution, a distribution over the unknown parameter λ_k . This prior distribution $f(\lambda_k)$ is then updated to the so-called posterior distribution $f(\lambda_k | \mathbf{t})$, the distribution over λ_k given the data \mathbf{t} , via Bayes' rule

$$f(\lambda_k | \mathbf{t}) \propto f(\mathbf{t} | \lambda_k) f(\lambda_k).$$

The posterior $f(\lambda_k | \mathbf{t})$ subsumes the information from both expert knowledge and data, and forms the basis for all inferences, like, e.g., predictions.

For the prior over λ_k , a convenient choice is to use the inverse Gamma distribution, which is commonly parametrized in terms of the parameters $a_k > 0$ and $b_k > 0$,

$$f(\lambda_k | a_k, b_k) = \frac{(b_k)^{a_k}}{\Gamma(a_k)} \lambda_k^{-a_k - 1} e^{-\frac{b_k}{\lambda_k}}, \quad (5)$$

in short, $\lambda_k \mid a_k, b_k \sim \text{IG}(a_k, b_k)$.

The inverse Gamma distribution is convenient because it is a conjugate prior, i.e., the posterior obtained by Bayes' rule is again an inverse Gamma distribution and thus easily tractable because only the prior parameters need to be updated to obtain the posterior parameters, so no numerical integration or simulation techniques are necessary. Furthermore, this conjugacy holds also when right-censored observations are used for updating, as indicated below.

Instead of the usual parametrization in terms of a_k and b_k , we use the parameters $n_k > 1$ and $y_k > 0$ because they are more easy to interpret. They are defined as

$$n_k = a_k - 1 \quad \text{and} \quad y_k = b_k/n_k, \quad (6)$$

where y_k can be interpreted as the prior guess for the scale parameter λ_k , as $E[\lambda_k \mid n_k, y_k] = y_k$. Using (4), we can thus translate an expert's statement of the expected component lifetime into a corresponding value for y_k . n_k can be seen as a prior strength or pseudocount, this will become clear in the discussion of the update step below.

The parametrization in terms of n_k and y_k also clarifies the nature of the combination of prior information and data through Bayes' rule. In the conjugate setting, applying Bayes' rule simply means that the prior parameters n_k and y_k are updated to posterior parameters, which we denote by n'_k and y'_k , respectively. Assume we observe $N_k = e_k + c_k$ component lifetimes, where e_k is the number of actual failure events, and c_k is the number of right-censored observations. We denote the failure times by $t_{k,1}, \dots, t_{k,e_k}$, and the censoring times by $t_{k,1}^+, \dots, t_{k,c_k}^+$, and collect them in an observation vector $\mathbf{t}_k = (t_{k,1}, \dots, t_{k,e_k}, t_{k,1}^+, \dots, t_{k,c_k}^+)$. Then, the updated, posterior parameters are

$$n'_k = n_k + e_k, \quad y'_k = \frac{n_k}{n_k + e_k} y_k + \frac{e_k}{n_k + e_k} \frac{\tau(\mathbf{t}_k)}{e_k}, \quad (7)$$

where $\tau(\mathbf{t}_k) = \sum_{i=1}^{e_k} (t_{k,i})^\beta + \sum_{i=1}^{c_k} (t_{k,i}^+)^\beta$.

From the simple update rule (7), we see that y'_k is a weighted average of the prior parameter y_k and the maximum likelihood estimator (ML) $\tau(\mathbf{t}_k)/e_k$, with weights proportional to n_k and e_k , respectively. n_k can thus be interpreted as a prior strength or pseudocount, indicating how much our prior guess should weigh against the e_k observed failure events. Furthermore, $\text{Var}[\lambda \mid n_k, y_k] = (y_k)^2/(1 - 1/n_k)$; for fixed y_k , a higher n_k indicates that more probability mass is concentrated around y_k .

Using (6) and (7), the posterior distribution over λ_k is given by

$$\lambda_k \mid n_k, y_k, \mathbf{t}_k \sim \text{IG}(n_k + e_k + 1, n_k y_k + \tau(\mathbf{t}_k)). \quad (8)$$

As this posterior can be defined in terms of the updated, posterior parameters n'_k and y'_k , we may also write

$$f(\lambda_k \mid n_k, y_k, \mathbf{t}_k) = f(\lambda_k \mid n'_k, y'_k).$$

The iterative nature of Bayesian inference means that we can take the updated, posterior parameters as new prior parameters and update them again using new data. We will use this property to use the extra information about components that accumulates during operation of the system: At any time $t_{\text{now}} > 0$, all non-failed components contribute a right censored observation t_{now}^+ ; any failed components instead contribute a fully observed, non-censored lifetime, and both types of observations can be used in the update step (7).

4.2. Component posterior predictive distributions determined at t_{now}

To calculate the current system reliability function $R_{\text{sys}}^{(t_{\text{now}})}(t)$, we need, for each $k = 1, \dots, K$, the probabilities $P(C_t^k = l_k)$ for the number of type k components that function at times $t > t_{\text{now}}$, taking into account all information available at t_{now} . In the Bayesian framework, these probabilities are given by a so-called posterior predictive distribution that can be derived from the posterior over λ .

Denote by $n_k^{(0)}$ and $y_k^{(0)}$ the parameters reflecting the knowledge base at system start-up time $t = 0$. These could have been obtained by updating prior parameters n_k and y_k (reflecting expert knowledge) to n_k' and y_k' using test data, or could be taken directly equal to n_k and y_k if no test data is available. In both cases, the distribution $f(\lambda_k | n_k^{(0)}, y_k^{(0)})$ thus accounts for all knowledge on component type k that is available at system start-up.

Following our comments at the end of Section 4.1, the component models can be further updated using information gained from the current operational cycle as of t_{now} : failure times of failed components, and the right-censoring time t_{now}^+ for each non-failed component, can be used to update $n_k^{(0)}$ and $y_k^{(0)}$ according to (7). We denote the resulting parameter values by $n_k^{(t_{\text{now}})}$ and $y_k^{(t_{\text{now}})}$.

In analogue to the notation used in Section 4.1, let $N_k = e_k^{(t_{\text{now}})} + c_k^{(t_{\text{now}})}$ be the number of type k components in the monitored system, where $e_k^{(t_{\text{now}})}$ denotes the number of type k components that have failed by t_{now} during the current operational cycle, and $c_k^{(t_{\text{now}})}$ denotes the number of type k components that have not failed by t_{now} during the current operational cycle. We can thus collect these observations, as of t_{now} , in a vector $\mathbf{t}_k^{(t_{\text{now}})} = (t_{k,1}, \dots, t_{k,e_k^{(t_{\text{now}})}}, t_{\text{now}}^+, \dots, t_{\text{now}}^+)$, containing $c_k^{(t_{\text{now}})}$ right-censored observations t_{now}^+ . The resulting posterior predictive distribution for any $t > t_{\text{now}}$ is obtained as

$$\begin{aligned} & P(C_t^k = l_k | n_k^{(0)}, y_k^{(0)}, \mathbf{t}_k^{(t_{\text{now}})}) \\ &= \binom{c_k^{(t_{\text{now}})}}{l_k} \int [P(T^k > t | T^k > t_{\text{now}}, \lambda_k)]^{l_k} \times \\ & \quad [P(T^k \leq t | T^k > t_{\text{now}}, \lambda_k)]^{c_k^{(t_{\text{now}})} - l_k} f(\lambda_k | n_k^{(0)}, y_k^{(0)}, \mathbf{t}_k^{(t_{\text{now}})}) d\lambda_k, \quad (9) \end{aligned}$$

where T^k is the Weibull distributed lifetime of a component of type k . Note that through the condition $T^k > t_{\text{now}}$, we also take into account that the components

in the system have the age t_{now} . Now, by the Weibull assumption (3), one has

$$\begin{aligned} P(T^k \leq t \mid T^k > t_{\text{now}}, \lambda_k) &= \frac{P(t_{\text{now}} < T^k \leq t \mid \lambda_k)}{P(T^k > t_{\text{now}} \mid \lambda_k)} \\ &= \frac{F(t \mid \lambda_k) - F(t_{\text{now}} \mid \lambda_k)}{1 - F(t_{\text{now}} \mid \lambda_k)} = 1 - e^{-\frac{t^{\beta_k} - (t_{\text{now}})^{\beta_k}}{\lambda_k}}. \end{aligned} \quad (10)$$

Consider the posterior (8) obtained using observations $\mathbf{t}_k^{(t_{\text{now}})}$, written in terms of the updated parameters $n_k^{(t_{\text{now}})}$ and $y_k^{(t_{\text{now}})}$. Substituting this posterior and (10) into (9), we get

$$\begin{aligned} P(C_t^k = l_k \mid n_k^{(0)}, y_k^{(0)}, \mathbf{t}_k^{(t_{\text{now}})}) &= \binom{c_k^{(t_{\text{now}})}}{l_k} \int \left[e^{-\frac{t^{\beta_k} - (t_{\text{now}})^{\beta_k}}{\lambda_k}} \right]^{l_k} \left[1 - e^{-\frac{t^{\beta_k} - (t_{\text{now}})^{\beta_k}}{\lambda_k}} \right]^{c_k^{(t_{\text{now}})} - l_k} \times \\ &\quad \frac{(n_k^{(t_{\text{now}})} y_k^{(t_{\text{now}})})^{n_k^{(t_{\text{now}})} + 1}}{\Gamma(n_k^{(t_{\text{now}})} + 1)} \lambda_k^{-(n_k^{(t_{\text{now}})} + 1) - 1} e^{-\frac{n_k^{(t_{\text{now}})} y_k^{(t_{\text{now}})}}{\lambda_k}} d\lambda_k \\ &= \binom{c_k^{(t_{\text{now}})}}{l_k} \sum_{j=0}^{c_k^{(t_{\text{now}})} - l_k} (-1)^j \binom{c_k^{(t_{\text{now}})} - l_k}{j} \frac{(n_k^{(t_{\text{now}})} y_k^{(t_{\text{now}})})^{n_k^{(t_{\text{now}})} + 1}}{\Gamma(n_k^{(t_{\text{now}})} + 1)} \times \\ &\quad \int \lambda_k^{-(n_k^{(t_{\text{now}})} + 1) - 1} \exp \left\{ -\frac{(l_k + j)(t^{\beta_k} - (t_{\text{now}})^{\beta_k}) + n_k^{(t_{\text{now}})} y_k^{(t_{\text{now}})}}{\lambda_k} \right\} d\lambda_k. \end{aligned} \quad (11)$$

The terms remaining under the integral form the core of an inverse Gamma distribution (5) with parameters $n_k^{(t_{\text{now}})} + 1$ and $n_k^{(t_{\text{now}})} y_k^{(t_{\text{now}})} + (l_k + j)(t^{\beta_k} - (t_{\text{now}})^{\beta_k})$, allowing to solve the integral using the corresponding normalization constant. We thus have, for $l_k \in \{0, 1, \dots, c_k^{(t_{\text{now}})}\}$,

$$\begin{aligned} P(C_t^k = l_k \mid n_k^{(0)}, y_k^{(0)}, \mathbf{t}_k^{(t_{\text{now}})}) &= \binom{c_k^{(t_{\text{now}})}}{l_k} \sum_{j=0}^{c_k^{(t_{\text{now}})} - l_k} (-1)^j \binom{c_k^{(t_{\text{now}})} - l_k}{j} \times \\ &\quad \left(\frac{n_k^{(t_{\text{now}})} y_k^{(t_{\text{now}})}}{n_k^{(t_{\text{now}})} y_k^{(t_{\text{now}})} + (l_k + j)(t^{\beta_k} - (t_{\text{now}})^{\beta_k})} \right)^{n_k^{(t_{\text{now}})} + 1}, \end{aligned} \quad (12)$$

$$\text{where } n_k^{(t_{\text{now}})} y_k^{(t_{\text{now}})} = n_k^{(0)} y_k^{(0)} + \sum_{i=1}^{e_k^{(t_{\text{now}})}} (t_{k,i})^{\beta_k} + c_k^{(t_{\text{now}})} (t_{\text{now}})^{\beta_k}$$

$$\text{and } n_k^{(t_{\text{now}})} = n_k^{(0)} + e_k^{(t_{\text{now}})}.$$

4.3. Conditional system reliability function determined at t_{now}

Now that we can calculate the component posterior predictive distributions at t_{now} , we can use the method from Section 3 to determine the dynamic system residual life distribution (RLD) $R_{\text{sys}}^{(t_{\text{now}})}(t)$, giving us the probability that the system functions at times $t > t_{\text{now}}$, taking into account all information available at t_{now} :

$$R_{\text{sys}}^{(t_{\text{now}})}(t) = \sum_{l_1=0}^{c_1^{(t_{\text{now}})}} \cdots \sum_{l_K=0}^{c_K^{(t_{\text{now}})}} \Phi^{(t_{\text{now}})}(l_1, \dots, l_K) \prod_{k=1}^K P(C_t^k = l_k \mid n_k^{(0)}, y_k^{(0)}, \mathbf{t}_k^{(t_{\text{now}})}). \quad (13)$$

Since this is the distribution of system reliability conditional on the system surviving until t_{now} , we have that $R_{\text{sys}}^{(t_{\text{now}})}(t_{\text{now}}) = 1$ for any t_{now} before system failure. Note that if one or several components have failed by t_{now} , the system reliability block diagram changes, and with it the survival signature $\Phi(l_1, \dots, l_K)$. We denote the current survival signature by $\Phi^{(t_{\text{now}})}(l_1, \dots, l_K)$.

For each prediction time t , (13) is a sum over $\prod_{k=1}^K (c_k^{(t_{\text{now}})} + 1)$ terms. However, some of these terms correspond to $\Phi^{(t_{\text{now}})}(l_1, \dots, l_K) = 0$, which can thus be disregarded. For each of the remaining terms, we must calculate the product $\prod_{k=1}^K P(C_t^k = l_k \mid n_k^{(0)}, y_k^{(0)}, \mathbf{t}_k^{(t_{\text{now}})})$. For all products, the constituting factors can be taken from the same table enumerating $P(C_t^k = l_k \mid n_k^{(0)}, y_k^{(0)}, \mathbf{t}_k^{(t_{\text{now}})})$ for all $l_k \in \{0, 1, c_k^{(t_{\text{now}})}\}$, $k = 1, \dots, K$, so (12) needs to be evaluated only $\sum_{k=1}^K (c_k^{(t_{\text{now}})} + 1)$ times.

We will evaluate (13) on a dense grid of prediction times $t > t_{\text{now}}$, thus discretely approximating the RLD. As the evaluation of (13) for each t does not involve any complex numeric calculations or Monte Carlo sampling, the grid of prediction values can be very fine.

5. Dynamic and adaptive maintenance policy based on the current system residual life distribution

Now we can derive the adaptive and dynamic maintenance policy that resembles a CBM policy. Our idea is described here first in general terms, and then illustrated along two exemplary timelines in Section 5.1. Afterwards, we give a detailed description of the policy in Section 5.2, and expand on the involved computations in Sections 5.3 and 5.4.

We denote by t_{now} the moment in time that the system is observed, measured from the moment the present operational cycle has started, which coincides with starting to use a new system. Instead of determining a threshold for a degradation signal like in usual CBM approaches, we determine, for t_{now} , the adaptive and dynamic time span $\tau_*^{(t_{\text{now}})}$ which gives the time from t_{now} until the next optimal moment to re-evaluate the system. A preventive maintenance action is initiated when $t_{\text{now}} + \tau_*^{(t_{\text{now}})}$ would be earlier than the next planned evaluation

time point. We describe this dynamic and adaptive maintenance policy in more detail in Section 5.2. The optimal time to system re-evaluation $\tau_*^{(t_{\text{now}})}$ is determined based on $R_{\text{sys}}^{(t_{\text{now}})}(t)$ by minimizing $g^{(t_{\text{now}})}(\tau)$, the expected cost rate for the current operational cycle. This criterion, as detailed in Section 5.3, only assumes that a RLD is given, and thus is not tied to our choice for the component models. One could also consider alternative optimality criteria to determine $\tau_*^{(t_{\text{now}})}$; when safety is paramount one could, e.g., determine $\tau_*^{(t_{\text{now}})}$ such that the probability of system failure is at most at some very low pre-determined level. Section 5.4 then focuses on our choice for the component models and the resulting form for $R_{\text{sys}}^{(t_{\text{now}})}(t)$, and shows how $\tau_*^{(t_{\text{now}})}$ is numerically determined in this situation.

5.1. Two exemplary timelines

Figure 2 shows two exemplary time lines, starting each with a new system (operational cycle) at $t = 0$. The timeline on the left leads to preventive maintenance, the timeline on the right to corrective maintenance.

As described in Section 1, the health of the system is re-evaluated at three types of triggers. Triggers of type (1) are the evenly spaced planned evaluation time points $m\delta$, where $m \in \mathbb{N}_0$. The times $m\delta$ are indicated by vertical tick lines on the time axis in Figure 2, where $\delta = 0.1$. The moments of failure of a component form the other two types of triggers. A component failure after which the system still functions is a trigger of type (2), depicted as a (non-filled) circle on the time axis. A filled circle indicates a trigger of type (3), which is a component failure that causes the whole system to fail.

For each trigger of type (1) and (2), the system health is evaluated by calculating $\tau_*^{(t_{\text{now}})}$, the currently optimal time to re-evaluation, visualized by the thick horizontal bars in Figure 2. For a type (3) trigger, the system health evaluation just corresponds to acknowledging that the system has failed. For triggers of type (1) and (2), the calculated $\tau_*^{(t_{\text{now}})}$ is compared to the time to the next planned evaluation time point. When $\tau_*^{(t_{\text{now}})}$ is shorter than this time, preventive maintenance of the system is initiated.

In the timeline on the left of Figure 2, $\tau_*^{(t_{\text{now}})}$ is re-evaluated at times $t_{\text{now}} = 0.1, 0.2, 0.24, 0.3, 0.36, 0.4$, where 0.24 and 0.36 correspond to component failure times. Note that the failure at $t_{\text{now}} = 0.24$ substantially shortens $\tau_*^{(0.24)}$, such that $0.24 + \tau_*^{(0.24)} < 0.2 + \tau_*^{(0.2)}$, i.e., the optimal moment of re-evaluation has moved backwards in time. This illustrates that new insights drawn from a component failure (regarding the reliability block diagram as well as the component model) may have considerable influence on the system RLD and the $\tau_*^{(t_{\text{now}})}$ that is derived from it. The component failure at $t_{\text{now}} = 0.36$ leads to a further shortened $\tau_*^{(0.36)}$, such that $\tau_*^{(0.36)} < 0.1 = \delta$. However, this does not lead to an immediate decision for preventive maintenance, since $\tau_*^{(0.36)}$ is compared to the time to the next planned evaluation at $t = 0.4$. The decision for preventive maintenance is instead made at $t_{\text{now}} = 0.4$, as only then it is not economical any more to wait for the next planned evaluation at $t = 0.5$.

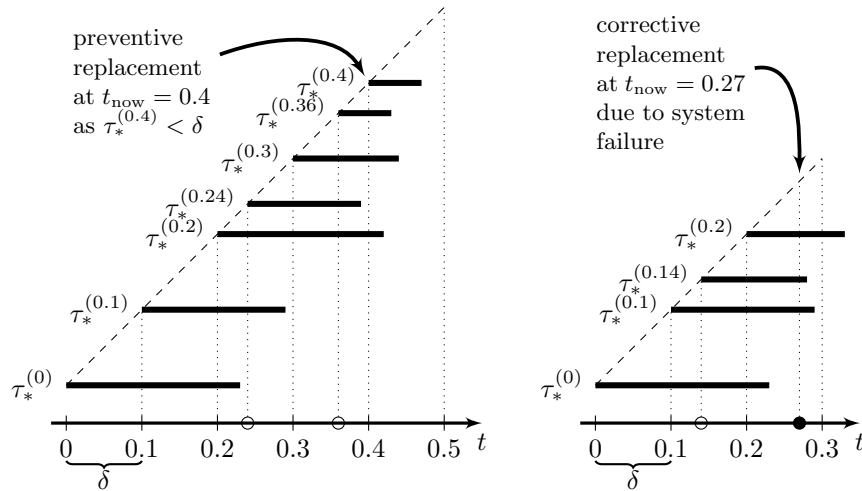


Figure 2: Two exemplary timelines for the suggested maintenance policy. The horizontal bars depict $\tau_*^{(t_{\text{now}})}$, the time to the next economically optimal evaluation of the system RLD, for subsequent times t_{now} . The times t_{now} for which $\tau_*^{(t_{\text{now}})}$ is evaluated are at regular intervals $m\delta, m \in \mathbb{N}_0$ (trigger 1), and at the time of component failures (trigger 2), which are marked by a circle on the time axis. Failures that lead to the whole system failing (trigger 3) are marked by a filled circle.

The timeline on the right of Figure 2 illustrates an operational cycle ending with corrective maintenance. Here, the component failure at $t_{\text{now}} = 0.27$ leads to system failure. The size of $\tau_*^{(0.2)}$ indicates that the component failure that happens at $t_{\text{now}} = 0.27$ had been deemed very unlikely at time $t_{\text{now}} = 0.2$, but then nevertheless occurred at $t = 0.27$.

5.2. Detailed description of the policy

Each operational cycle starts with a system startup. At this point in time, the system is in an (as good as) new state, i.e., all components in the system are (as good as) new and have not aged. An operational cycle ends when the system is either maintained preventively, or the system fails, which triggers a corrective maintenance action.

We assume that both preventive and corrective maintenance result in an (as good as) new system state by replacing all components, such that the system begins a new operational cycle always under the same conditions. At first sight, replacing a complete system when only some parts of it are defective may seem strange. However, there are many real life examples, including from daily live. So called complete clustering (Olde Keizer et al., 2017) may be optimal in case of a very strong economic dependence between the different parts of a system, e.g., due to a very high fixed set-up cost independent of the maintenance activities to be done, as is the case of installations used in the generation and distribution of oil and gas. Note that often for safety reasons, the latter installations are

monitored continuously. There may be also technical dependencies between parts of a system that may require combined replacement: replacing the cassette of a bike often also requires to replace the chain, and the tires of an airplane are required to have the same thickness. But also complete systems are replaced because replacing one or more of its parts may take much longer, which may result in big production losses. Another reason can be that there may be a high probability that when individual parts are replaced, other parts will be damaged, requiring their replacement as well.

Replacing not all components during maintenance would require to account for the different ages of components in the system at the start of a new cycle. While this is in principle possible in our approach, exploring such selective replacement schemes is out of scope for this paper. Optimizing over replacement schemes is, in our view, a highly interesting topic that merits an investigation of its own. We will return to this in the conclusion (Section 8).

Figure 3 depicts the flow chart for the policy, indicating the decisions that are repeatedly encountered in a single operational cycle. The flow chart also contains the loop back to the start of a new operational cycle.

Consider a grid of planned evaluation time points $\{m\delta : m \in \mathbb{N}_0\}$, with δ a sufficiently small time increment. During an operational cycle, $\tau_*^{(t_{\text{now}})}$ is usually recalculated at times $t_{\text{now}} = m\delta$, $m = 0, 1, 2, \dots$. However, if a component i of type k fails before the next t_{now} on the grid, t_{now} is set instead to the exact failure time $t_{k,i}$. Next, we check whether the system has failed due to this component failure. If the system has failed, a corrective maintenance action is carried out, incurring cost c_c . If the system has not failed, we calculate $\tau_*^{(t_{\text{now}})}$ for the current off-grid time $t_{\text{now}} = t_{k,i}$, and move back onto the grid afterwards, unless there is another component failure before the next planned evaluation time.

The calculation of $\tau_*^{(t_{\text{now}})}$ for a fixed t_{now} resembles the calculation of the maintenance interval in an age based policy by trading off the costs of preventive and corrective maintenance (see Section 5.3 below). However, because $\tau_*^{(t_{\text{now}})}$ is derived from the current $R_{\text{sys}}^{(t_{\text{now}})}(t)$, our policy is adaptive and dynamic like a CBM policy. When computation of the RLD is fast, frequent recalculation of $\tau_*^{(t_{\text{now}})}$ is feasible, and it is useful because, even in absence of failures of system components, all components age, and also non-failure adds information to the component models, reducing uncertainty.

Once $\tau_*^{(t_{\text{now}})}$ is calculated, it is compared to the time span from the current t_{now} to the next planned evaluation time. If the current t_{now} is a time on the grid, then $t_{\text{now}} = m\delta$ for some $m \in \mathbb{N}_0$. The next planned evaluation is then at $(m+1)\delta$, and we compare $\tau_*^{(t_{\text{now}})}$ to δ . If $\tau_*^{(t_{\text{now}})} < \delta$, a preventive maintenance action is triggered, incurring cost c_p , otherwise we attempt to move to the next planned evaluation point, by setting $m = (m+1)$.

If the current t_{now} is instead an off-grid time corresponding to failure of component i of type k at time $t_{k,i}$, after which the system still functions, then the next planned evaluation time is the next time on the grid. Let m be such that $m\delta < t_{k,i} < (m+1)\delta$, so the next time on the grid is $(m+1)\delta$, and we

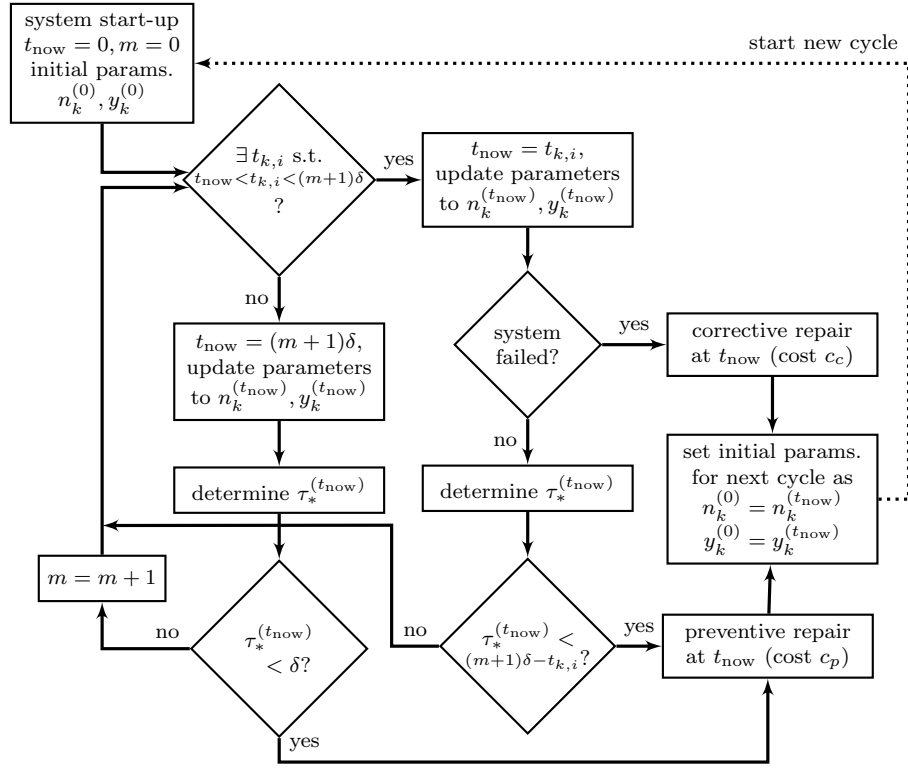


Figure 3: Flowchart for the proposed maintenance policy, showing a single operational cycle, including the loop back to system startup that starts a new cycle as a dashed arrow.

compare $\tau_*^{(t_{\text{now}})}$ with $(m+1)\delta - t_{k,i}$. If $\tau_*^{(t_{\text{now}})} < (m+1)\delta - t_{k,i}$, a preventive maintenance action is triggered, incurring cost c_p , otherwise we attempt to move to the next planned evaluation point, which is still $(m+1)\delta$.

In both cases, we decide to maintain the system preventively when the optimal time to re-evaluation is closer to now than the next time we intend to re-evaluate the system RLD, since otherwise, in waiting to the next planned evaluation time, one would exceed the optimal system failure risk as determined by the trade-off made via $g^{(t_{\text{now}})}(\cdot)$. When $\tau_*^{(t_{\text{now}})}$ is instead further in the future than the next planned evaluation time, we can safely continue to the next planned evaluation time, without risking to miss the cost-optimal moment to preventively maintain the system.

At the end of each operational cycle, i.e., after maintaining the system either preventively or correctively, the component information gained during the completed operational cycle is added to the component knowledge base by updating the parameters n_k and y_k accordingly, such that $n_k^{(0)}$ and $y_k^{(0)}$ for the next operational cycle account for the (censored) lifetimes observed during the previous operational cycles.

5.3. Expected operational cycle cost rate

To determine the optimal time to maintenance $\tau_*^{(t_{\text{now}})}$, we use the expected one-cycle cost rate (Ansell et al., 1984; Mazzuchi and Soyer, 1996; Coolen-Schrijner and Coolen, 2006) as the unit cost rate. As discussed in the literature review (Section 2), we cannot use the renewal theory based way to calculate the unit cost rate as often encountered in CBM literature (e.g., Si et al., 2013; Kim et al., 2011).

Let c_p be the cost of preventive maintenance, and c_c the cost of corrective (unplanned, breakdown) maintenance, where $c_p < c_c$. Usually, set-up costs, the actual maintenance costs (labor, new components) and downtime costs are much higher for corrective maintenance, such that often, $c_p \ll c_c$.

Let $T_{\text{sys}}^{(t_{\text{now}})}$ be the random variable corresponding to $R_{\text{sys}}^{(t_{\text{now}})}(t)$, giving the (random) system failure time for times $t > t_{\text{now}}$, conditioned on all information available at t_{now} , and $f_{\text{sys}}^{(t_{\text{now}})}(t)$ be the corresponding probability density. Note that due to the conditional nature of $R_{\text{sys}}^{(t_{\text{now}})}(t)$, which gives the probability that the system functions at time t given that it functions at t_{now} , also $f_{\text{sys}}^{(t_{\text{now}})}(t)$ is a conditional probability density, with support $\{t \in \mathbb{R} : t > t_{\text{now}}\}$.

Conditional on the realization $t_{\text{sys}}^{(t_{\text{now}})}$ of $T_{\text{sys}}^{(t_{\text{now}})}$, the unit cost rate for planning maintenance at τ time units after t_{now} is

$$g(\tau | T_{\text{sys}}^{(t_{\text{now}})} = t_{\text{sys}}^{(t_{\text{now}})}) = \begin{cases} c_p / (t_{\text{now}} + \tau) & \text{if } t_{\text{sys}}^{(t_{\text{now}})} \geq t_{\text{now}} + \tau \\ c_c / t_{\text{sys}}^{(t_{\text{now}})} & \text{if } t_{\text{now}} < t_{\text{sys}}^{(t_{\text{now}})} < t_{\text{now}} + \tau. \end{cases} \quad (14)$$

Note that here τ is a time span starting from t_{now} , and so is a time on a prospective time scale, where 0 corresponds to t_{now} . In contrast, t_{now} and $t_{\text{sys}}^{(t_{\text{now}})}$ are times on an absolute time scale, where 0 corresponds to the time of a new system start-up. Taking the expectation over $T_{\text{sys}}^{(t_{\text{now}})}$ in (14) leads to the expected operational cycle cost rate

$$\begin{aligned} g^{(t_{\text{now}})}(\tau) &= \mathbb{E}[g(\tau | T_{\text{sys}}^{(t_{\text{now}})})] \\ &= \frac{c_p}{t_{\text{now}} + \tau} R_{\text{sys}}^{(t_{\text{now}})}(t_{\text{now}} + \tau) + c_c \int_0^\tau \frac{1}{t_{\text{now}} + \theta} f_{\text{sys}}^{(t_{\text{now}})}(t_{\text{now}} + \theta) d\theta. \end{aligned} \quad (15)$$

Here, we assume that the time required to execute preventive and corrective maintenance can be neglected when compared with the duration of one operational cycle. $g^{(t_{\text{now}})}(\tau)$ gives the cost rate that we expect to incur over the complete operational cycle when, at time t_{now} , we schedule preventive maintenance for time $t_{\text{now}} + \tau$, so $g^{(t_{\text{now}})}(\tau)$ trades off the risk of system failure before $t_{\text{now}} + \tau$ with the gains we could reap when the system survives until $t_{\text{now}} + \tau$. Figure 4 shows $R_{\text{sys}}^{(t_{\text{now}})}(t)$ and the corresponding $g^{(t_{\text{now}})}(\tau)$ for a number of current times t_{now} in a numeric example which is described in more detail in Section 6.

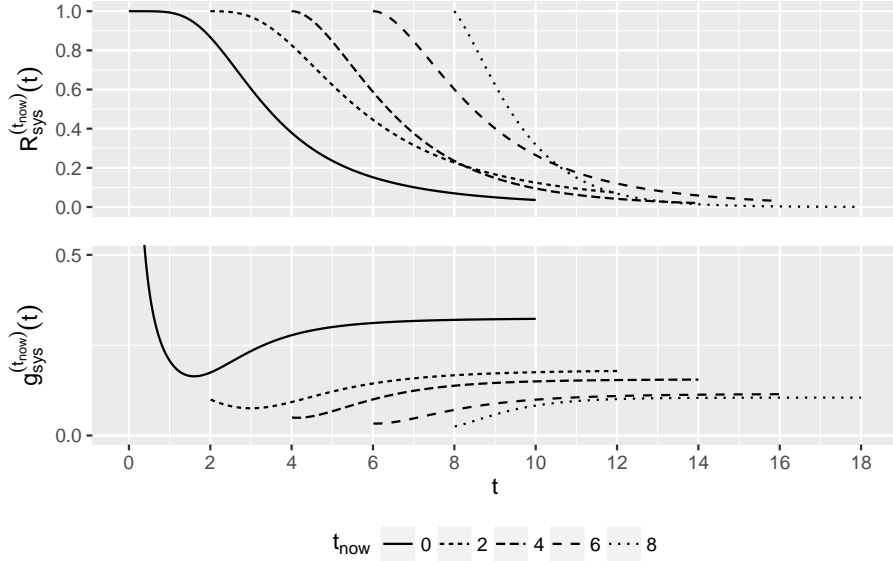


Figure 4: System reliability and unit cost rate functions for $t_{\text{now}} = 0, 2, 4, 6, 8$ for the system depicted in Figure 1 and prior component models as given in Table 2, assuming the failure history $P_2 = 3, P_3 = 4, C_2 = 6, C_3 = 7, H = 8$, and costs $c_u = 1, c_p = 0.2$.

Note that for the integral in (15) to exist, $E[1/(T_{\text{sys}}^{(t_{\text{now}})} - t_{\text{now}})]$ must exist, which is a condition for $f_{\text{sys}}^{(t_{\text{now}})}(t_{\text{now}} + \theta)$ for values of θ close to 0 (Coolen-Schrijner and Coolen, 2006). For our approach, however, this condition is not relevant, as we use a discrete approximation of $R_{\text{sys}}^{(t_{\text{now}})}(t)$ and $f_{\text{sys}}^{(t_{\text{now}})}(t)$, respectively (see Section 5.4 below).

$\tau_*^{(t_{\text{now}})} := \arg \min g^{(t_{\text{now}})}(\tau)$ is the cost-optimal time to do maintenance as of t_{now} , incorporating all that is known at current time t_{now} . The minimal expected operational cycle cost rate $g_*^{(t_{\text{now}})} := g^{(t_{\text{now}})}(\tau_*^{(t_{\text{now}})})$ corresponding to $\tau_*^{(t_{\text{now}})}$ thus is, as of t_{now} , the lowest cost rate for the current operational cycle that we can expect to attain. $g_*^{(t_{\text{now}})}$ generally decreases with t_{now} since for larger t_{now} , maintenance costs are spread over a longer time period. However, when many (or few but critical) components fail, $g_*^{(t_{\text{now}})}$ may increase, since then, system failure becomes much more likely, leading to a higher risk of incurring the corrective maintenance cost c_c . Figure 5 illustrates these relations for the scenario depicted in Figure 4. As can be seen there and in the numerical examples in Section 6, for continuous component lifetime distributions, $\tau_*^{(t_{\text{now}})}$ is a smooth function of t_{now} during periods where no component fails, with discrete jumps at component failure times. The size of the jump depends on the relevance of the failed component for the reliability of the system at the moment that the component failure occurs.

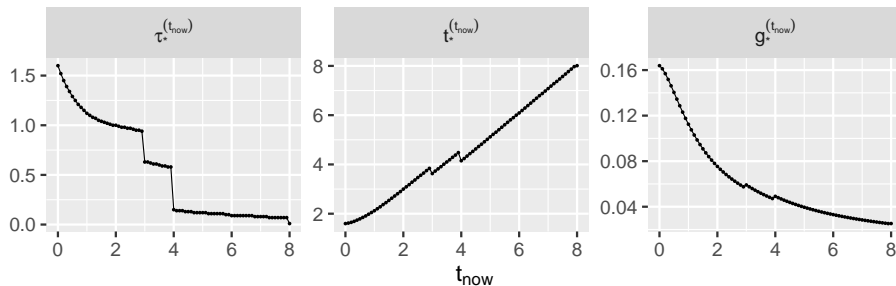


Figure 5: $\tau_*^{(t_{\text{now}})}$, $t_*^{(t_{\text{now}})} = t_{\text{now}} + \tau_*^{(t_{\text{now}})}$ and $g_*^{(t_{\text{now}})} = g^{(t_{\text{now}})}(\tau_*^{(t_{\text{now}})})$ for the system depicted in Figure 1, assuming the failure history $P_2 = 3$, $P_3 = 4$, $C_2 = 6$, $C_3 = 7$, $H = 8$, and costs $c_u = 1$, $c_p = 0.2$.

Note that common cause failures, i.e., failures of several components at the same time due to a shared root cause, are currently not accounted for in our model. Such simultaneous failures will seriously influence the RLD and thus $\tau_*^{(t_{\text{now}})}$. We will return to this in the conclusion (Section 8).

5.4. Numerical optimization of the expected operational cycle cost rate

For our choice of the component models, the resulting $R_{\text{sys}}^{(t_{\text{now}})}(t)$ as of (13) is closed-form, but not tractable analytically. Therefore, we evaluate $R_{\text{sys}}^{(t_{\text{now}})}(t)$ on a dense grid of time points t covering the time span from t_{now} to some horizon time $t_{\text{now}} + t_h$. We calculate $g^{(t_{\text{now}})}(\tau)$ on the same grid, and determine $\tau_*^{(t_{\text{now}})}$ by minimizing over this grid. Here, we approximate the integral in (15) numerically by a sum, where $f_{\text{sys}}^{(t_{\text{now}})}(t)$ is calculated via differences of $R_{\text{sys}}^{(t_{\text{now}})}(t)$.

The horizon time span t_h can be determined by practical considerations, but generally should reach far enough into the tail of $R_{\text{sys}}^{(t_{\text{now}})}(t)$ such that $g^{(t_{\text{now}})}(\tau)$ is not cut off before it reaches its minimum. Note that according to (15), the calculation of $g^{(t_{\text{now}})}(\tau)$ requires evaluations of $R_{\text{sys}}^{(t_{\text{now}})}(t)$ and $f_{\text{sys}}^{(t_{\text{now}})}(t)$ only for $t_{\text{now}} < t \leq t_{\text{now}} + \tau$. The choice of t_h is thus not critical, in the sense that $g^{(t_{\text{now}})}(\tau)$ evaluations will be correct even if $t_h < \tau_*^{(t_{\text{now}})}$.

We have implemented our method in R (R Core Team, 2016), using the package `ReliabilityTheory` (Aslett, 2016) to calculate survival signatures (code available on request). As the parameter update step is closed-form, recalculation of $\tau_*^{(t_{\text{now}})}$ is a matter of seconds, allowing for a near real-time update.

6. Further elaboration of the case study

We now explain in more detail the case study used for the previous illustrative figures (Figures 4 and 5). We use the simplified automotive brake system as introduced in Section 3, see Figure 1 for the reliability block diagram.

The cost parameters are chosen as $c_c = 1$, $c_p = 0.2$, i.e., corrective maintenance is five times more costly than preventive maintenance. This makes it

k	β_k	$E[T_i^k]$	$y_k^{(0)}$	$n_k^{(0)}$
C	2	3	11.5	1
H	1	10	10.0	1
M	2.5	8	244.1	1
P	1.5	5	13.0	1

Table 2: Prior parameters for the four component types as used in the numerical examples.

worthwhile to aim for a preventive repair of the system, but corrective maintenance will not be avoided at all cost.

The choice of prior parameters for the four component models are discussed in Section 6.1. Then, our maintenance policy is illustrated under three different exemplary failure time histories, described in Sections 6.2, 6.3 and 6.4.

6.1. Prior component models

The prior component model parameters chosen for this case study are given in Table 2. Note that the scale parameter for component type H is 1, such that the failure times for this component type are assumed to follow an Exponential distribution. In cases where only H and one or both of P_3 and P_4 are functioning, the system reliability function will thus be close to exponential, making preventive maintenance unattractive.

Note also that $n_k^{(0)} = 1$ for all four component types, indicating very weak prior knowledge. Observed failure times thus have a strong influence on the updating of $y_k^{(0)}$, such that component models are primarily data-driven. This offers the advantage of quick adaption for the case that an expert's assessment of the component MTTF turns out to be inadequate. When $n_k^{(0)}$ and $y_k^{(0)}$ also include test data, then $n_k^{(0)}$ will typically be larger, since the update of n_k involves the number of observed failures (see Section 4.1).

The expected component failure behaviour obtained by the parameter choices in Table 2 is visualized in Figure 6, showing the prior predictive reliability function for a single component of each type. Note that these functions are not plain Weibull reliability functions, but weighted averages over Weibull reliability functions according to the prior distribution over the scale parameter λ_k . Judging the adequacy of the prior parameter choices thus requires the sampling model to be correct. Methods for checking the sampling model are discussed, e.g., in Bayarri and Berger (2000). Figure 6 indicates that, e.g., the median prior expected failure times are 2.2, 4.1, 6.3, and 3.1 for type C , H , M , and P components, respectively. While it is a priori quite unlikely that any type C component will survive past time 6, we expect 25% of type H components to survive past time 10.

6.2. Failure history 1: failures later than expected

In the first failure history, we observe $P_2 = 3$, $P_3 = 4$, $C_2 = 6$, $C_3 = 7$ and $H = 8$, and in following the system until $t_{\text{now}} = 8$, all other failure times are

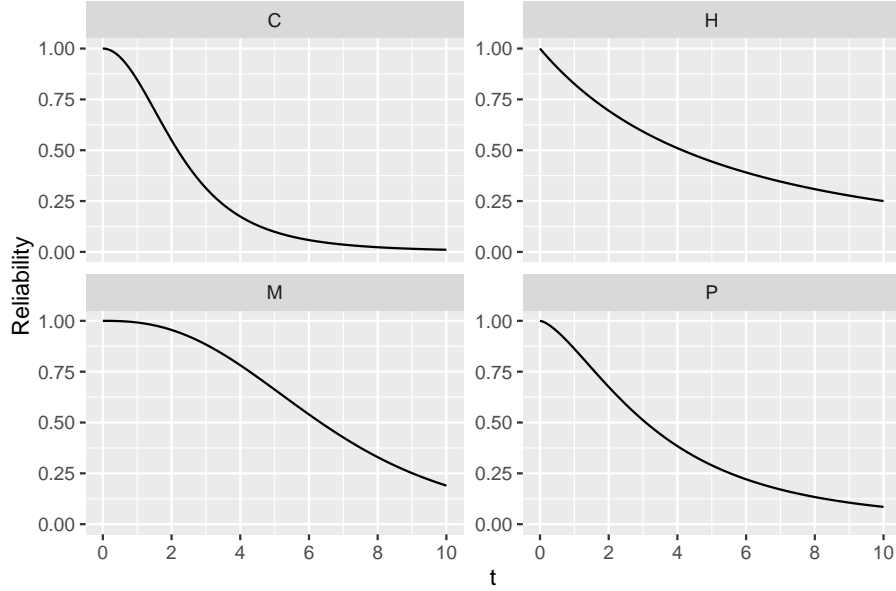


Figure 6: Prior predictive reliability functions corresponding to the values from Table 2, showing the expected failure behaviour.

right-censored. This is the failure history already used for Figures 4 and 5. Here, failures observed for type C are thus considerably later than expected, while for type P , the two observed failures are close to the assumed MTTF. Both failure at time 8 for H and non-failure by time 8 for M are nothing unusual. In total, due to the late type C failures, prior assumptions turn out to be somewhat pessimistic.

This is reflected in Figure 5, where $\tau_*^{(t_{\text{now}})}$ (left panel) first decreases quickly but then tends to decrease more slowly during failure-free periods, which is a consequence of adjusting prior expectations for type C components due to their unexpected reliability. Large drops of $\tau_*^{(t_{\text{now}})}$ happen only at the failures of components P_2 , P_3 and H at times 3, 4 and 8, respectively, as $R_{\text{sys}}^{(t_{\text{now}})}(t)$ changes considerably at these moments. $\tau_*^{(t_{\text{now}})}$ falls below $\delta = 0.1$ at time $t_{\text{now}} = 6$, triggering preventive maintenance of the system. $\tau_*^{(t_{\text{now}})} = 1.60$ at $t_{\text{now}} = 0$, this corresponds to the optimal maintenance time in an age-based policy that takes the system as monolithic. Since $\tau_*^{(t_{\text{now}})}$ does not change dramatically most of the time, $t_*^{(t_{\text{now}})}$, the cost-optimal moment of maintenance on the absolute timescale (center panel), steadily increases, with small drops corresponding to drops in $\tau_*^{(t_{\text{now}})}$, showing that as time progresses, accumulated information indicates that maintenance can be postponed. Consequently, $g_*^{(t_{\text{now}})}$ (right panel) is steadily decreasing, with small upward jumps at the failures of P_2 and P_3 . At $t_{\text{now}} = 6$, the expected operational cycle cost rate is 0.033, and so considerably lower

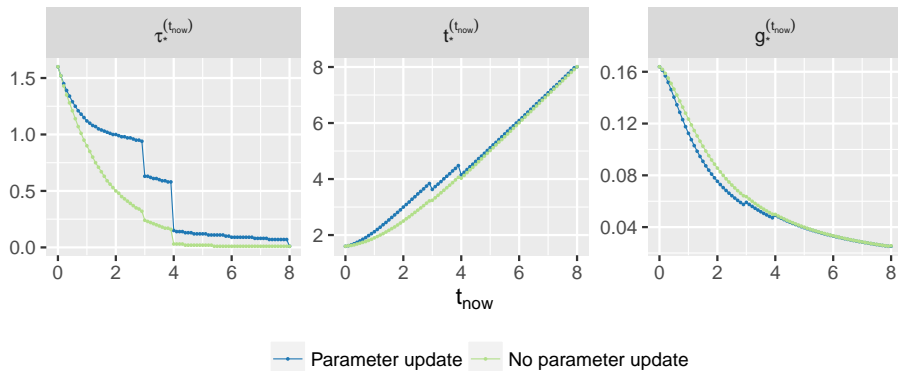


Figure 7: Illustration of the gains due to the parameter update at each t_{now} , assuming the failure history $P_2 = 3, P_3 = 4, C_2 = 6, C_3 = 7, H = 8$.

than the operational cycle cost rate of $0.2/1.6 = 0.125$ that one obtains by maintaining the system preventively at time 1.6, as suggested by the age-based policy that disregards the information from component monitoring. In this example, using the component failure information allows to lower the cost rate by 73%.

The effect of the continuous parameter update in our policy is illustrated in Figure 7. There, the results as seen already in Figure 5 are compared to the case when component parameters are not updated during the operation of the system, while still accounting for the fact that components age, and that a failed component changes the system layout. We see that $\tau_*^{(t_{\text{now}})}$ (left panel) is decreasing more quickly in the no update case, with the jumps at component failure times being less pronounced than in the update case. Without the continuous update, residual system lifetime continues to be underestimated, and maintenance is scheduled too early, leading to higher expected operational cycle cost rates $g_*^{(t_{\text{now}})}$ (right panel). $\tau_*^{(t_{\text{now}})}$ for the policy without parameter update drops below $\delta = 0.1$ at $t_{\text{now}} = 4$, while maintenance is triggered for the algorithm with parameter update at $t_{\text{now}} = 6$. The parameter update thus allows to run the system for two more time units, corresponding to a unit cost rate of $0.2/6 = 0.033$ instead of $0.2/4 = 0.05$, a reduction of 33%.

6.3. Failure history 2: failure times as expected

In a second failure history, we observe the failure times $C_2 = 1, C_2 = 3, P_2 = 0.5, P_3 = 1.5$, and follow the system until $t_{\text{now}} = 2.5$. Here, components of type C behave more or less as expected, while type P components fail slightly earlier than expected. Overall, prior assumptions are reasonably in line with observed failure behaviour. Consequently, the difference between the continuous update case and the no update case is small, as visualized in Figure 8. There, $\tau_*^{(t_{\text{now}})}$ values for the policy with parameter updates vary around those obtained

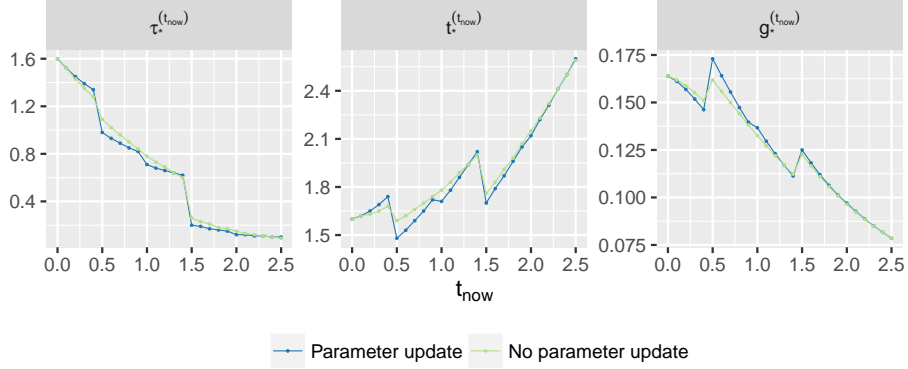


Figure 8: Effect of the continuous parameter update for the second failure history ($C_2 = 1$, $C_2 = 3$, $P_2 = 0.5$, $P_3 = 1.5$) which is more or less in line with expectations. With observed failure behaviour more or less in line with expectations, both the policies with and without continuous update perform similarly, unlike in case of the first failure history as depicted in Figure 7, where updating gave considerable gains.

for the policy without parameter updates, and a similar behavior for $t_*^{(t_{\text{now}})}$ and $g_*^{(t_{\text{now}})}$ is observed. With $\delta = 0.1$, preventive maintenance is carried out at times 2.6 and 2.5 for the two policies, respectively. This shows that for this failure history, in which there is not much to be gained by the parameter update, $\tau_*^{(t_{\text{now}})}$ behaviour is dominated by the update of the system reliability block diagram. Here, any differences due to the update are very small in comparison to the gains offered by the continuous update in case of the first failure history (note the differing vertical and horizontal scales in Figures 7 and 8).

6.4. Failure history 3: failures earlier than expected

The third failure history contains very early failures with $C_2 = 0.1$, $C_3 = 0.2$, $C_4 = 0.8$, $H = 1$, $P_2 = 0.3$, $P_3 = 0.4$, and we follow the system until $t_{\text{now}} = 1$, see Figure 9.

The policy without parameter update leads to consistently higher $\tau_*^{(t_{\text{now}})}$ values, while the policy with parameter update takes the surprisingly early failures into account. Unlike in the other two failure histories, $t_*^{(t_{\text{now}})}$ is generally decreasing, indicating that maintenance should happen much earlier than expected initially. The policy with parameter update consistently suggests to maintain earlier than the policy without update, but for both models, $\tau_*^{(t_{\text{now}})}$ drops below $\delta = 0.1$ upon the failure of H at $t_{\text{now}} = 1$, obtaining an operational cycle cost rate of 0.2. In contrast, the age-based policy that disregards the information from component monitoring would have scheduled maintenance for time 1.6, but it is very unlikely that the system will still function by that time. If the system fails between time 1 and time 1.6, then the age-based policy would lead to an operational cycle cost of 0.625 to 1, much higher than for our policy.

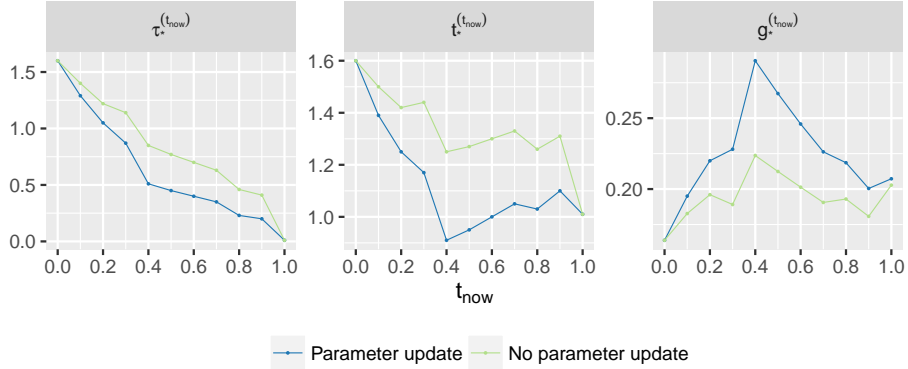


Figure 9: Effect of the continuous parameter update for the third failure history, containing very early failures ($C_2 = 0.1$, $C_3 = 0.2$, $C_4 = 0.8$ $H = 1$, $P_2 = 0.3$, $P_3 = 0.4$).

Like in failure history 2, the structural information (which components still function?) seems to deliver a major part of the gains for our CBM policy, whereas the parameter update seems most effective in situations where component failures happen later than expected. These findings are confirmed by results from a small simulation study described next.

7. A small simulation study

We conducted a small simulation study to illustrate the performance of our policy as compared to corrective and age-based policies. To be able to discern the effect of parameter learning over several operational cycles, we simulated component failure times for five subsequent operational cycles for each simulation repetition. We thus can compare policies that use information gained in previous operational cycles with policies that do not update component parameters at the end of an operational cycle. We also distinguish condition-based maintenance policies with and without continuous parameter update as in Section 6. In total, we compare the six maintenance policies listed in Table 3, which also gives the abbreviations used in Figures 10, 11 and 12.

We compare these policies by three simple performance measures, calculated for each repetition. e_{sys} is the number of system failures that occurred when applying the policy, This is an integer between zero and five, as there are five operational cycles per repetition. \bar{r}_{sys} is the mean realized system runtime, i.e., the average of the realized operational cycle length, taken over the five operational cycles per repetition. \bar{g} is the mean realized operational cycle cost rate obtained over the five operational cycles. This is calculated as the sum of preventive and corrective maintenance costs incurred in the five operational cycles, divided by the sum of the five realized operational cycle lengths.

Policy	Description
CBM-cpu	Condition-based maintenance (continuous parameter update)
CBM-epu	Condition-based maintenance (end of cycle parameter update)
CBM-npu	Condition-based maintenance (no parameter update)
ABM-epu	Age-based maintenance (end of cycle parameter update)
ABM-npu	Age-based maintenance (no parameter update)
CM	Corrective maintenance

Table 3: The six maintenance policies compared in the simulation study.

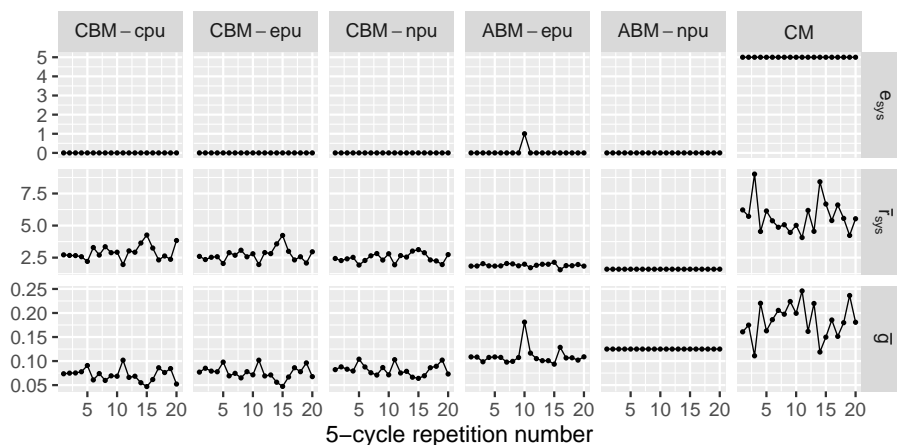


Figure 10: Simulation study with 20 repetitions, where each repetition consists of 5 subsequent operational cycles. Case A: $\lambda_k = y_k^{(0)}$.

We use again the simplified automotive brake system depicted in Figure 1, together with the prior parameters for the component models as given in Table 2, and assume that $c_c = 1$ and $c_p = 0.2$ like before. We study three data scenarios A, B and C, corresponding to failure times as expected, as well as failure times earlier and later than expected.

7.1. Case A: failure times as expected, $\lambda_k = y_k^{(0)}$

For this scenario, we simulated component failure times according to prior expectations, i.e., component failure times were simulated as Weibull failure times with the scale parameter λ_k chosen as equal to the expert's prior guess $y_k^{(0)}$ for all four component types. The results are depicted in Figure 10.

Obviously, the corrective maintenance policy (CM) results in $e_{\text{sys}} = 5$ for all 20 repetitions, as it cannot lead to preventive maintenance. In contrast, all condition-based (CBM) and age-based policies can avoid system failures entirely (with a single exception for ABM-epu). Mean system runtimes are longest for CM because it runs the system until failure, but this leads to the highest mean

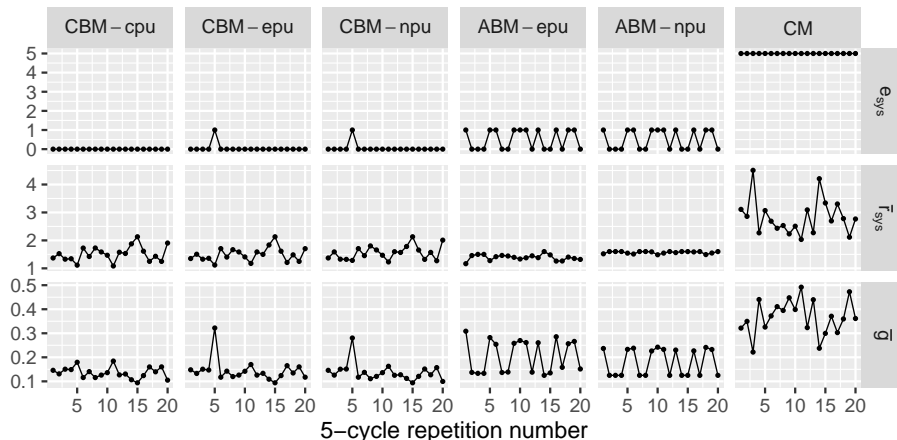


Figure 11: Simulation study with 20 repetitions, where each repetition consists of 5 subsequent operational cycles. Case B: $\lambda_k = 0.5y_k^{(0)}$.

cost rates. The condition-based maintenance policy with continuous parameter update (CBM-cpu) obtains the lowest mean cost rates, since it leads to the longest system runtimes without system failures. CBM-epu and CBM-npu have slightly higher cost rates, as they lead to shorter system runtimes. The age-based policy without parameter update (ABM-npu) always schedules maintenance for $\tau_*^{(0)} = 1.6$. As the system does not fail before this time in any of the five operational cycles within the 20 repetitions, the mean realized system runtime is constantly 1.6, with mean cost rate $0.2/1.6 = 0.125$.

7.2. Case B: failure times earlier than expected, $\lambda_k = 0.5y_k^{(0)}$

For the second scenario, component failure times were simulated as being earlier than expected, by choosing λ_k for all component types as half the size of the expert's prior guess, i.e., $\lambda_k = 0.5y_k^{(0)}$ for all $k = C, H, M, P$. The results are depicted in Figure 11.

Here, only CBM-cpu can avoid system failures entirely. Both CBM-epu and CBM-npu result in one system failure in the $20 \times 5 = 100$ system runs, whereas the two age-based policies each result in ten system failures. System runtimes are less variable for the age-based policies since they do not use the current system structure information, i.e., which components have failed and which not. System runtimes are more variable for the condition-based policies, but longer on average, leading to lower realized cost rates. Again, CBM-cpu performs slightly better than CBM-epu as it has the possibility to learn within an operational cycle as well, managing to avoid system failures entirely. Still, CBM-npu does surprisingly well, considering that it never updates prior component parameters. This confirms the observation from Section 6.4 that current system structure information delivers a major part of the gains for our CBM policy.

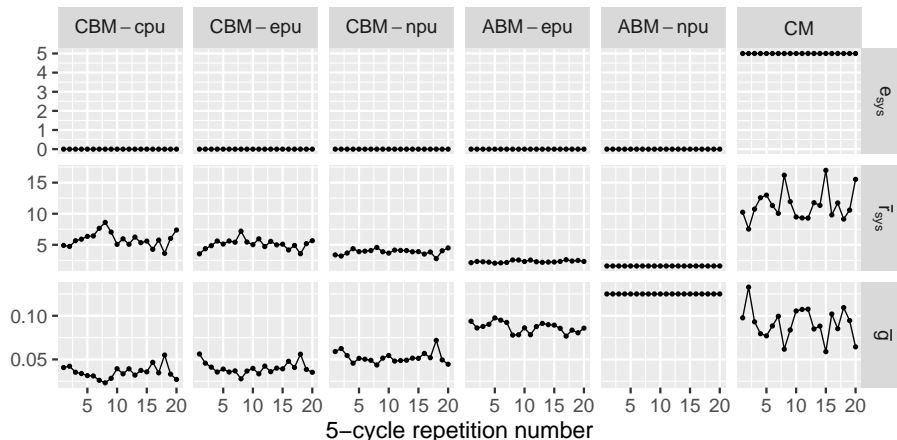


Figure 12: Simulation study with 20 repetitions, where each repetition consists of 5 subsequent operational cycles. Case C: $\lambda_k = 2y_k^{(0)}$.

7.3. Case C: failure times later than expected, $\lambda_k = 2y_k^{(0)}$

The third scenario consists of component failure times that are later than expected, obtained by simulating Weibull lifetimes using $\lambda_k = 2y_k^{(0)}$ for all four component types. The results are depicted in Figure 12.

As the expert’s information now underestimates component lifetimes, all policies (except the corrective policy) act quite cautiously, leading to complete avoidance of system failures. However, the mean system runtimes \bar{r}_{sys} exhibit a clear hierarchy between age-based and condition-based policies, and within these two groups a clear effect of the parameter update strategy. Here, CBM-cpu obtains the longest system runtimes, and thus the lowest cost rates, with a clear advantage over CBM-epu and CBM-npu.

Overall, it seems that CBM-cpu can exert its adaptive strengths most effectively when prior assumptions are too pessimistic, while also performing well when prior assumptions are correct or too optimistic.

8. Summary and outlook

In this paper, we have proposed a new condition-based maintenance policy for systems based on monitoring of the functioning of components. We use the survival signature to compute system reliability functions for given component models, and propose to use Weibull component models with conjugate priors to express expert knowledge. The resulting policy shows promising results for different numerical examples and a small simulation study.

There are several aspects in our modeling that warrant further investigation. In this paper, we assume that component failure times are observed precisely via monitoring. However, in many applications, the component status can only be

determined at inspections, leading to interval-censored failure time observations. Extending the model to account for interval-censored failure times would allow to derive a condition-based maintenance policy with wider-spaced inspections by determining the optimal moment for the next inspection of the system.

Another way to relax the assumption of precise component monitoring is to consider unreliable detection of component failures. Our approach may be quite sensitive to the non-detection of component failures, and one could tackle this by including a probabilistic observation model, which assigns probabilities to false positives and false negatives in the component monitoring.

We also assume that the component-specific Weibull shape parameters are known, and update only the scale parameters. We restricted ourselves to this simplistic modelling to allow for a clear and straight-forward demonstration of the general approach. In a practical application, the shape parameters should be jointly updated together with the scale parameters. This could be implemented by adopting the discretized approach by Soland (1969).

As mentioned at the end of Section 5.3, we currently disregard the possibility of common cause failures, i.e., simultaneous failures with a shared root cause. Common cause failures can drastically reduce the system reliability by leading to sudden system failures. To offer a realistic estimation of the optimal time to maintenance $\tau_*^{(t_{\text{now}})}$, the system RLD $R_{\text{sys}}(t)$ needs to account for such failures when they are possible. When common cause failures can affect several component types at once, the joint posterior predictive probabilities $P\left(\bigcap_{k=1}^K \{C_t^k = l_k\}\right)$ in (1) cannot be obtained as a product of component posterior predictive probabilities any more. We think that by combining ideas from Coolen and Coolen-Maturi (2015) and Troffaes et al. (2014), the proposed policy could be extended to accommodate common cause failures.

In our policy, we assumed that in both preventive and corrective replacement all components are restored to an as good as new state. An obvious alternative is to only replace the failed components, but would this offer any advantage? Furthermore, could it be even optimal to replace a subset of the failed components only? Such selective component replacement schemes seem to be a promising area for research. To account for different ages of components at system startup, one could introduce artificial component types in (1), grouping components both by type and by age. However, computational complexity increases with the number of component types, so it seems most promising to replace batches of components, or all components of one type.

A further simplifying assumption relates to the cost of preventive and corrective replacements, which we consider as fixed. In many real-life applications, maintenance costs vary over time (e.g., increased labour cost due to overtime or night shift premiums), and may depend on the number of failed components, or even on the exact system state. In our approach, time-dependent cost structures can be easily accounted for, by letting c_p and c_c in (14) depend on τ . It will be more complex to account for costs that are dependent on the system state, but our approach already provides the necessary information in form of the posterior predictive distributions derived in Section 4.2.

Another promising extension would be to use sets of conjugate priors for the component models. This approach is described in Walter and Coolen (2017), leading to sets of residual life distributions $R_{\text{sys}}^{(t_{\text{now}})}(t)$. Consequently, we would obtain a set of expected operational cycle cost rate functions $g^{(t_{\text{now}})}(\tau)$. The determination of the (set of) optimal time(s) to maintenance $\tau_*^{(t_{\text{now}})}$ is then non-trivial, and would involve decision criteria discussed in the imprecise probability literature, like E-admissibility, maximality, or maximin (see, e.g., §8 Augustin et al., 2014).

Acknowledgments

Gero Walter was supported by the DINALOG project CAMPI (“Coordinated Advanced Maintenance and Logistics Planning for the Process Industries”). Both authors would like to thank Frank Coolen for inspiring discussions and valuable hints on maintenance optimization criteria.

Bibliography

- Ahmadzadeh, F., Lundberg, J., 2014. Remaining useful life estimation: review. *International Journal of System Assurance Engineering and Management* 5, 461–474, <http://link.springer.com/article/10.1007/s13198-013-0195-0>.
- Ansell, J., Bendell, A., Humble, S., 1984. Age replacement under alternative cost criteria. *Management Science* 30, 358–367, <http://www.jstor.org/stable/2631808>.
- Aslett, L. J. M., 2016. ReliabilityTheory: Tools for structural reliability analysis. R package, <http://www.louisaslett.com>.
- Augustin, T., Coolen, F. P. A., de Cooman, G., Troffaes, M. C. M., 2014. *Introduction to Imprecise Probabilities*. Wiley, New York.
- Barlow, R., Proschan, F., 1975. *Statistical Theory of Reliability and Life Testing*. Holt, Rinehart and Winston, Inc., New York.
- Bayarri, M. J., Berger, J. O., 2000. P values for composite null models. *Journal of the American Statistical Association* 95 (452), 1127–1142, <http://dx.doi.org/10.1080/01621459.2000.10474309>.
- Coolen, F. P. A., Coolen-Maturi, T., 2012. Generalizing the signature to systems with multiple types of components. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (Eds.), *Complex Systems and Dependability*. Vol. 170 of *Advances in Intelligent and Soft Computing*. Springer, pp. 115–130, http://dx.doi.org/10.1007/978-3-642-30662-4_8.

- Coolen, F. P. A., Coolen-Maturi, T., 2015. Predictive inference for system reliability after common-cause component failures. *Reliability Engineering and System Safety* 135, 27–33, <http://dx.doi.org/10.1016/j.res.2014.11.005>.
- Coolen-Schrijner, P., Coolen, F. P. A., 2006. On optimality criteria for age replacement. *Journal of Risk and Reliability* 220, 21–28, <http://dx.doi.org/10.1243/1748006XJRR1>.
- Kim, M. J., Jiang, R., Makis, V., Lee, C.-G., 2011. Optimal Bayesian fault prediction scheme for a partially observable system subject to random failure. *European Journal of Operational Research* 214, 331–339, <http://dx.doi.org/10.1016/j.ejor.2011.04.023>.
- Mazzuchi, T. A., Soyer, R., 1996. A Bayesian perspective on some replacement strategies. *Reliability Engineering and System Safety* 51, 295–303, [http://dx.doi.org/10.1016/0951-8320\(95\)00077-1](http://dx.doi.org/10.1016/0951-8320(95)00077-1).
- Olde Keizer, M. C. A., Flapper, S. D. P., Teunter, R. H., 2017. Condition-based maintenance policies for systems with multiple dependent components: A review. *European Journal of Operational Research* 261, 405–420, <http://dx.doi.org/10.1016/j.ejor.2017.02.044>.
- R Core Team, 2016. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, <https://www.R-project.org/>.
- Reed, S., 2017. An efficient algorithm for exact computation of system and survival signatures using binary decision diagrams. *Reliability Engineering and System Safety* 165, 257–267, <http://dx.doi.org/10.1016/j.res.2017.03.036>.
- Si, X.-S., Wang, W., Chen, M.-Y., Hu, C.-H., Zhou, D.-H., 2013. A degradation path-dependent approach for remaining useful life estimation with an exact and closed-form solution. *European Journal of Operational Research* 226, 53–66, <http://dx.doi.org/10.1016/j.ejor.2012.10.030>.
- 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, 1–14, <http://dx.doi.org/10.1016/j.ejor.2010.11.018>.
- Soland, R. M., 1969. Bayesian analysis of the Weibull process with unknown scale and shape parameters. *IEEE Transactions on Reliability* 18, 181–184, <http://dx.doi.org/10.1109/TR.1969.5216348>.
- Sun, J., Zuo, H., Wang, W., Precht, M. G., 2012. Application of a state space modeling technique to system prognostics based on a health index for condition-based maintenance. *Mechanical Systems and Signal Processing* 28, 585–596, <http://dx.doi.org/10.1016/j.ymsp.2011.09.029>.

Troffaes, M. C. M., Walter, G., Kelly, D. L., 2014. A robust Bayesian approach to modelling epistemic uncertainty in common-cause failure models. *Reliability Engineering and System Safety* 125, 13–21, <http://dx.doi.org/10.1016/j.ress.2013.05.022>.

Walter, G., Coolen, F. P. A., 2017. Robust Bayesian reliability for complex systems under prior-data conflict. Conditionally accepted for *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, submission available at <http://arxiv.org/abs/1610.07222>.