An Alert-Assisted Inspection Policy for a Production Process with Imperfect Condition Signals

Citation for published version (APA):

Document license:
CC BY

DOI:
10.1016/j.ejor.2021.05.051

Document status and date:
Published: 16/04/2022

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher’s website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 07. Apr. 2022
Production, Manufacturing, Transportation and Logistics

An alert-assisted inspection policy for a production process with imperfect condition signals

Alp Akcay

Department of Industrial Engineering and Innovation Sciences, Eindhoven University of Technology, Postbus 513, Eindhoven 5600 MB, the Netherlands

A R T I C L E   I N F O

Article history:
Received 8 October 2020
Accepted 28 May 2021
Available online 4 June 2021

Keywords:
Maintenance
Inspection planning
Imperfect predictive alerts
Delay-time model
Dynamic programming

A B S T R A C T

We study the inspection scheduling decisions for a production process that goes through a hidden defective state before its failure. The production process is equipped with a predictive model, generating alert and no-alert signals. An alert signal indicates that production process is in the defective state, while a no-alert signal indicates it is in the healthy state. The signals are imperfect, meaning that an alert signal can be generated for a healthy process and a no-alert signal can be generated for a defective process. Only a costly inspection can detect the true condition. We introduce a new inspection policy, which generalizes the age-based inspection policy that performs planned inspections at predetermined intervals, by considering that an inspection can also be triggered by a certain number of alerts from the predictive model. To optimize the proposed inspection policy, a stochastic dynamic programming model is formulated with the objective of minimizing the long-run expected cost rate. The performance improvement achieved by the optimal policy is quantified by comparing it to practically relevant benchmark policies. Numerical experiments with a set of realistic problem instances show that adding alert-triggered inspections to traditional age-based inspection scheduling brings up to 44% reduction in the expected cost rate when the predictive model is sufficiently accurate. Characterizing the performance of the optimal policy at a given level of imperfectness is especially useful in practice as it allows making an assessment on how much can be invested to justify a certain level of improvement in the predictive model.

© 2021 The Author(s). Published by Elsevier B.V.
This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

1. Introduction

Data-driven models are becoming popular for predicting the condition of industrial systems. Examples include building a random-forest classifier to generate predictive alerts for signaling the defective condition of a machine tool (Wang, 2017) and for predicting the condition of a medical-grade monitor in an X-ray scanner (van Lohuizen, 2018). However, the predictions from data-driven models can be incorrect (i.e., due to limitations in data quality, model misspecification, or prediction errors), leading to a wrong conclusion on the true condition of the asset. In such a case, it is unlikely to completely replace conventional inspection policies (e.g., inspections at predetermined intervals) with purely prediction-driven inspection policies. Finding the right balance between these two is key to obtain the most benefit from data-driven models.

The motivation behind this work comes from our collaboration with Philips Consumer Lifestyle (PCL), a global consumer-electronics manufacturer. PCL produces electric shavers by using highly precise electrochemical machining tools. These tools are subject to random failures, and before a tool fails, it goes through a defective phase where the tool can continue processing new products (see Akcay, Topan, and van Houtum 2021 for details). Only a physical inspection can exactly detect the true condition of the tool. Recently, a data-driven prediction model has been developed to generate an alert when there is an indication of tool failure (Wang, 2017). It then becomes a practical question to answer how to react to these alerts given that they are not always correct.

In the current practice of PCL, a tool is inspected as soon as a specific number of products has been processed (Akcay et al., 2021; Maarse, 2017). This is essentially a periodic-inspection policy and can be considered as an extreme because the inspection is triggered only by a fixed amount of usage and any information on the real-time condition of the tool is ignored. The other extreme is the inspect-at-first-alert policy, which means not to keep track of the usage but perform an inspection as soon as there is an alert from the predictive model. This could work well if the alert and no-alert signals are highly accurate. However, these signals are imperfect, meaning that an alert signal can be raised for a healthy tool (i.e., false positive) while it is possible to observe a no-alert signal...
signal for a defective tool (i.e., false negative). Thus, a good inspection policy requires the right balance between trusting the alerts from the predictive model and still performing inspections at predetermined intervals.

In order to achieve this balance, we introduce a new inspection policy referred to as the (s, u)-policy. In a production process with a hidden operational state (i.e., the process can be in a healthy or defective state, but this is not observed directly) and with a defect-prediction model generating an alert if the state of the process is estimated as defective, the (s, u)-policy works as follows: for a production process at age \( \tau \), an inspection is scheduled to be performed as soon as the process completes \( u(\tau) \) additional products or \( s(\tau) \) number of alerts is observed, whichever occurs first. Notice that the (s, u)-policy is very general. As special cases, it reduces to the periodic inspection policy (i.e., when \( s(\tau) = \infty \) and \( u(\tau) \) is fixed, for all \( \tau \)), the age-based inspection policy of Maillard and Pollock (2002) with aperiodic inspection intervals (i.e., when \( s(\tau) = \infty \) for all \( \tau \)), and the inspect-at-first-alert policy (i.e., when \( s(\tau) = 1 \) and \( u(\tau) = \infty \) for all \( \tau \)). The (s, u)-policy aims to strike a balance between performing an inspection not too early and not too late given the imperfectness of the alert and no-alert signals. Although our problem is motivated from a specific company, it captures a more general situation where imperfect information is periodically revealed about a system’s true condition, and only costly inspections can identify this true condition with certainty.

In this paper, we specifically answer the following research questions: (i) How to characterize the long-run expected-cost rate under the (s, u)-policy and solve for the optimal value of \( s(\tau) \) and \( u(\tau) \) at each possible process age \( \tau \)? (ii) When is it most beneficial to add an alert-triggered inspection option to conventional age-based inspection scheduling at predetermined intervals? When is it sufficient to only have planned inspections at predetermined intervals or to only have alert-driven inspections? (iii) What is the economic value of a better defect-prediction model? In particular, what is the reduction in the expected cost rate by improving the false-positive rate and the false-negative rate of the defect-prediction model?

The contributions of our paper can be summarized as follows. To find the optimal (s, u)-policy we formulate a stochastic dynamic programming model with the objective of minimizing the long-run expected cost rate by considering that the number of products processed in the healthy and defective states have general probability distributions. As an input for this formulation, the alert-arrival process is characterized in the presence of a defect-prediction model that generates imperfect alert and no-alert signals. To the best of our knowledge, the (s, u)-policy is new and has not been studied in the literature before. With the increasing popularity of data-driven prediction models for condition monitoring, the (s, u)-policy is especially attractive when the aim is to support traditional age-based inspection policies with the imperfect signals from such prediction models. It is often argued that smart maintenance requires blending traditional approaches with new tools and techniques, and the two need to work in a complementary fashion (Close, 2017). This is the intuition behind the (s, u)-policy in its effort to combine a traditional age-based inspection policy with alert-driven inspections. Through numerical examples along with sensitivity analysis, the performance improvement achieved by the optimal (s, u)-policy is quantified by comparing it with benchmark policies. By evaluating the performance of the optimal (s, u)-policy, it can also be answered how much can be further invested for reducing the imperfectness of the alert and no-alert signals (e.g., whether it is worth to invest in new sensors for collecting more data or put more effort for model development, etc.)

The remainder of this paper is organized as follows. Section 2 reviews the relevant literature and Section 3 formally introduces the inspection planning problem. Section 4 analyzes the alert arrival process for a defect-prediction model with a specific false-positive and false-negative probability. Section 5 characterizes the long-run expected cost rate under the (s, u)-policy and formulates the problem of finding the optimal (s, u)-policy as a stochastic dynamic program. Section 6 discusses how to solve this dynamic program, and Section 7 presents the numerical experiments and insights. Section 8 concludes the paper with future research directions.

2. Literature review

There is a rich literature on maintenance optimization covering various areas such as system replacement, inspections and repair (Jardine & Tsang, 2013; de Jonge & Scarf, 2020). Our paper complements the literature on inspection planning (e.g., Chehbi & Daoud 2009) by addressing the question of when to perform an inspection to detect the true condition of a system. In particular, there are three literature streams that are the most relevant to our work: (1) Age-based inspection scheduling under delay-time model, (2) planning of imperfect inspections, and (3) maintenance planning under imperfect condition signals. In the remainder of this section, each group is discussed in detail.

Age-based inspection scheduling under delay-time model

The deterioration model of the production process considered in our paper is also referred to as the delay-time model. There is a rich literature on maintenance planning under delay-time models (Christer, 1982). We refer the reader to Wang (2012) and Section 3.2 of de Jonge and Scarf (2020) for a review of this literature. Our work comes closest to Maillard and Pollock (2002), which optimize the aperiodic inspection intervals of an industrial system as a function of the system age with the objective of minimizing the average cost rate in the long run. Various extensions of the standard delay-time model have been studied. MacPherson and Glazebrook (2011) extend (Maillard & Pollock, 2002) by adding a choice between a cheap repair and a more expensive renewal of the system, should it be found to be in the defective state by an inspection. Postponement of preventive action in defective state is considered by van Oosterom, Elwany, Çelebi, and van Houtum (2014). Wang, Wang, and Peng (2017) consider a delayed first inspection and then periodic inspections, Cavalcante, Lopes, and Scarf (2018) consider an heterogeneous population of components with periodic inspections, and Scarf, Cavalcante, and Lopes (2019) consider random inspections (i.e., inspections arriving as a Poisson process and inspections that are periodic but randomly carried out). However, all the studies mentioned above consider perfect inspections (i.e., an inspection accurately reveals the true condition of the system) in the absence of condition monitoring or alert signals. In this work, we also consider perfect inspections but assume that condition signals are available in the form of imperfect alert and no-alert signals providing information on the hidden operational state of the system.

Planning of imperfect inspections

Due to the limitations of current technologies and/or human errors, imperfect inspections (i.e., failing to detect an existing problem or incorrectly reporting a problem) are quite common in practice. Imperfect inspection policies are widely studied under the delay-time degradation model; see Okumura, Jardine, and Yamashina (1996) for an early example. While we consider perfect inspections in our paper, the outcome of a defect-prediction model can be considered as performing an imperfect inspection at no cost, establishing the link with our paper and this literature stream. Under the delay-time model, the imperfectness of the inspections are often modelled as independent parameters representing the false-positive (i.e., labeling a healthy system as defective) and false-negative (i.e., labeling a defective system as healthy).
healthy) probabilities; e.g., Berrade, Scarf, Cavalcante, and Dwight (2013). Several extensions of imperfect inspections under delay-time model have been studied. Flage (2014) and Alberti, Cavalcante, Scarf, and Silva (2018) consider failure-inducing inspections. Berrade, Scarf, and Cavalcante (2017) consider the postponement option of preventive action for a defective system, and Cavalcante, Scarf, and Berrade (2019) consider silent failures. Driessen, Peng, and van Houtum (2017) and Zhang, Shen, and Ma (2020) allow the false-positive and false-negative probabilities to change over time. The objective in these studies is to optimize the inspection interval by using renewal theory with the objective of minimizing average cost-rate in the long run. Yang, Ye, Lee, Yang, and Peng (2019) and Zhang, Shen, Liao, and Ma (2021) consider two-phase maintenance policies (where there are different inspection policies in each phase). Levitin, Finkelstein, and Huang (2019) consider reliability-critical systems that perform missions of fixed durations and optimize the schedule of imperfect inspections with the objective of maximizing the mission completion probability at a given constraint on the probability of failure. Similar to the papers mentioned above, the decision in our paper is also about when to inspect a system. However, there is a fundamental difference. In our case, imperfect inspections are performed automatically (i.e., in a regular manner as long as the system operates) via a prediction model without any cost, and the information obtained from the prediction model is used for scheduling perfect inspections.

Maintenance planning under imperfect condition signals

Our paper comes closest to the literature which considers the availability of periodically obtained imperfect monitoring information (also referred to as inaccurate condition signals) in order to decide when to perform inspections. For a single system with a degradation model represented with a Markov chain, many structural properties have been derived by formulating the problem as a Partially Observable Markov Decision Process (POMDP) model; e.g., Ohnishi, Kawai, and Mine (1986), Özekici and Pliska (1991), and Grosfeld-Nir (1996). Kim and Makis (2013) analyze the joint optimization of sampling and maintenance decisions in the POMDP framework, and determine that the optimal information sampling and maintenance optimization policy consists of at most four regions. Also using a POMDP model, van Oosterom, Maillart, and Kharoufeh (2017) consider that periodically obtained sensor information can become worse over time, and address the problem of adaptively scheduling perfect inspections and sensor replacements. More recently, van Staden and Boute (2020) consider the effect of access to different quality levels for deterioration data, resulting in imperfect information about the underlying state of the system being monitored. The authors investigate the decision maker’s willingness to pay for the information that provides more accurate representation of the real condition. In the presence of an imperfect defect-prediction model, Xiao, Ackay, Maillart, and van Houtum (2020) analytically characterize when the structure of the optimal policy can be represented with a critical threshold on a belief variable that represents the likelihood that the system is in the healthy state. In the papers with a POMDP framework, it is often the case that a decision is made at every time period, e.g., inspect or skip the inspection, conditional on the information available on the true state of the system, which is summarized in a continuous belief state. On the other hand, motivated from the practical problem at PCL, we focus on the scheduling decision (i.e., when to do the next inspection). The scheduling decision is made at the moments when the true state of the system is known with certainty, making it unnecessary to maintain a belief state. To optimize the scheduling decisions, we build a stochastic dynamic programming model with the current age of the process as the state information. This brings us the following modeling advantages compared to a POMDP framework. In our paper, the healthy and defective states of the system are also hidden, but the degradation states do not necessarily evolve according to a Markov chain: we assume the duration of the healthy state and the defective state can follow a general probability distribution. Furthermore, the resulting policy (i.e., counting the number of alerts and the usage of the process, and comparing them with known thresholds to trigger an inspection) is practically more appealing than maintaining a continuous belief state and responding to it in a short time window. In terms of modeling methodology, our paper can be considered closest to MacPherson and Glazebrook (2014), which assume the availability of imperfect sensor information and optimizes the inspection schedule that minimizes the long-run average cost rate under a delay-time model. However, MacPherson and Glazebrook (2014) assume that if a transition to defective state is undetected by a sensor, there is no further opportunity to detect the defective state. While this can be a fine assumption for a sensor, it is hardly applicable when a data-driven prediction model is used for monitoring purpose.

3. Model formulation

In this section, the inspection planning problem under the (s, u)-policy will be formally introduced. Sections 3.1 and 3.2 introduce the failure model of the production process and the details of the defect-prediction model, respectively. Section 3.3 describes the (s, u)-policy, and Section 3.4 specifies the objective to consider in optimizing this policy. Table 1 provides a summary of all the mathematical notations to be used in the model formulation.

3.1. Deterioration model of the production process

We consider a production process that can be in one of the three states: healthy, defective, and failed. A healthy process goes into a defective state before it fails. The defective state is not visible from outside, and only an inspection can reveal it with certainty. Let \( X \in \mathbb{N}_0 \) denote the number of products processed by a healthy process until it becomes defective (e.g., \( X = 5 \) means that the process completes five products in healthy state, but the process becomes defective during the sixth product), where \( \mathbb{N}_0 = \{0, 1, 2, \ldots\} \) is the set of nonnegative integers. We let \( H \in \mathbb{N}_0 \) denote the number of products processed in the defective state (e.g., \( H = 2 \) means that the process completes two products in the defective state and fails during the processing of the next product). The probability mass function (pmf) and cumulative distribution function (cdf) of \( X \) are denoted with \( f_X() \) and \( F_X() \), respectively. Likewise, the pmf and cdf of \( H \) are denoted with \( f_H() \) and \( F_H() \). This so-called delay-time model is a commonly used deterioration process in the inspection-based maintenance literature (see, e.g., Baker & Christer, 1994; Driessen et al., 2017; van Oosterom et al., 2014; Wang, 2012; Yang et al., 2019 for examples in continuous-time setting and Ackay et al., 2021 for an example in discrete-time setting).

3.2. Defect-prediction model

We assume that a defect-prediction model is available to detect whether the production process has already reached the defective state. Such a predictive model can be regarded as a binary classifier: it generates an alert if the process is predicted as defective. If the process is predicted as healthy, then no alert is generated. The predictions can be imperfect with a false-positive probability \( \alpha \) and false-negative probability \( \beta \). To be specific, the defect-prediction model gives an alert for a healthy process with probability \( \alpha \), and it fails to give an alert for a defective process with probability \( \beta \). This is summarized in Table 2.

The training of a binary classifier requires minimizing the discrepancy between the model output and the actual condition by
Furthermore, the alert is important if it is generated before triggering an inspection for a process at age \( \tau \). 

### Table 1

| \( T \) | Total number of products to be processed |
| \( \tau \) | Age of the process |
| \( u(\tau) \) | The number of products to be processed until the next planned inspection for a process at age \( \tau \) |
| \( s(\tau) \) | The number of alerts to wait before triggering an inspection for a process at age \( \tau \) |
| \( X \) | The number of products processed by a healthy process until the process becomes defective |

\[ f_x(h) \] 

\[ f_n(h) \] 

\[ H \] 

\[ X \] 

\( f(T(x)) \) 

\( T(x) \) 

\( p_{\alpha}^\beta(t) \) 

\( I_p \) 

\( C_I \) 

\( C_F \) 

\( C_J \) 

\( g(T, \tau|\tilde{\epsilon}, \tilde{\delta}) \) 

\( N' \) 

\( \gamma' \) 

\( \beta \) 

\( \alpha \) 

The random variable that represents the number of products at which the \( i \)th alert (starting from the current age) is generated given that the process is currently healthy and it completes \( x \) more products in the healthy state.

\[ \text{Cost of an alert-triggered inspection} \]

\[ \text{Cost of a planned inspection} \]

\[ \text{Cost of processing a product in the defective state} \]

\[ \text{Cost of preventive maintenance} \]

\[ \text{Cost of corrective maintenance} \]

The expected total cost incurred in the next \( T \) products for a process at age \( \tau \) if the actions \( s(\tau) = \tilde{\alpha} \) and \( u(\tau) = \tilde{\delta} \) are taken at age \( \tau \) and the optimal \((s, u)\)-policy is followed thereafter.

\[ \gamma' \text{ Long-run expected cost per product under the optimal} \]

\( \text{(s, u)-policy} \)

<table>
<thead>
<tr>
<th>Prediction output</th>
<th>No alert</th>
<th>Alert</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>True state</strong></td>
<td><strong>Healthy</strong></td>
<td><strong>Defective</strong></td>
</tr>
<tr>
<td><strong>Defective</strong></td>
<td>True negative ((1 - \alpha))</td>
<td>False positive ((\alpha))</td>
</tr>
<tr>
<td></td>
<td>False negative ((\beta))</td>
<td>True positive ((1 - \beta))</td>
</tr>
</tbody>
</table>

### Table 2

Confusion matrix of the defect-prediction model.

The model parameters according to a specified measure of the prediction performance, e.g., precision, recall, F1-score (Florian, Sgarbossa, & Zennaro, 2021). Once the trained prediction model is validated by using a separate test data, the proportion of the alert instances among the healthy-process instances of the test data can be used as the false-positive probability \( \alpha \). Likewise, the proportion of the no-alert instances among the defective-process instances of the test data can be used as the false-negative probability \( \beta \). It is important to note that \( \alpha + \beta = 1 \) implies that the alert and no-alert signals come from a random classifier (Fawcett, 2006), which means predicting the process as defective at a fixed percentage of the time without exploiting any information from available data. Furthermore, if \( \alpha + \beta > 1 \) it is possible to negate the classifier (i.e., reverse its classification decisions on every instance) to achieve \( \alpha + \beta < 1 \).

The cost of generating an output from the prediction model is negligible. This is reasonable when the prediction model is already built and there is sufficient computing capacity available to run it. We assume that the defect-prediction model is run upon the completion of every product unless there is a failure or there is already a scheduled inspection. This reflects the situation that the prediction model is run per product by using the data related to the process specifications obtained during the manufacturing of that particular product. Running a prediction model at discrete times is not uncommon in practice, e.g., when the prediction model is run after collecting a new batch of data every hour, day, week, etc. Also, a widely performed data processing step is to batch or aggregate sensor data before running a prediction model, further justifying the generation of prediction model output at discrete time points.

For a given true condition of the process, the predictions are generated independently from each other; e.g., for a process which is in the healthy state, an alert is generated with probability \( \alpha \) upon the completion of a product, independent of the previous prediction outputs.

### 3.3. Description of the \((s, u)\)-policy

The idea behind the \((s, u)\)-policy is to support a classical age-based inspection policy (e.g., MacPherson & Glazebrook, 2011; Maillart & Pollock, 2002; Sabouri, Shechter, & Huh, 2015) with alert signals. Specifically, in addition to scheduled inspection moments, alerts from the defect-prediction model can also trigger an inspection. The characteristics of the production process and the details of the \((s, u)\)-policy are described as follows:

- There is a cost associated with continuing production with a defective process; e.g., the economic loss due to lower quality of the products. Let \( C_p \) denote the cost of processing a product in the defective state of the production process.
- The age of the process increases by one unit when the processing of a product is completed without failure. If the process fails, a corrective maintenance is performed at cost \( C_F \) and the next product starts with a healthy process at age zero.
- For a new process at age \( \tau = 0 \), or for a process which is inspected at age \( \tau > 0 \) and found healthy, the next inspection is scheduled to be performed after \( u(\tau) \) additional products or as soon as the \( s(\tau) \)th alert (starting from age \( \tau \)) is observed, whichever occurs first. The former is referred to as a planned inspection, and the latter is referred to as an alert-triggered inspection. In contrast to a defect-prediction model, inspections perfectly reveal the true state of the production process. The costs of performing an alert-driven inspection and a planned inspection are denoted with \( I_p \) and \( I_F \), respectively. Inspections do not alter the condition of the process.
- If an inspection finds that the process is defective, a preventive maintenance is performed at cost \( C_F \) and the processing of the next product starts with a healthy process at age zero. This implies the process is certainly in the healthy state after an inspection is performed. Therefore, the accumulated number of alerts is reset to zero after an inspection.
- If the processing of a product is completed without a failure, the defect-prediction model is run to see whether an alert is generated. However, if there is already an inspection planned at the end of that product, then the planned inspection is performed without running the defect-prediction model. Fig. 1 summarizes the \((s, u)\)-policy with a graphical illustration.

We refer to the inspection policy described above as the \((s, u)\)-policy. Notice that \( s(\tau) = 1 \) and \( u(\tau) = \infty \), \( \forall \tau \), imply performing an inspection if and only if there is an alert observed; i.e., inspect-at-first-alert policy. On the other hand, if \( s(\tau) \geq u(\tau) \), \( \forall \tau \), it means that an alert-driven inspection is never performed (because the \( s(\tau) \)th alert can be raised earliest by the completion of the \( s(\tau) \)th product, but \( s(\tau) \geq u(\tau) \) implies there is already a planned inspection) and the \((s, u)\)-policy becomes equivalent to the age-based inspection policy with predetermined (possibly aperiodic) inspection intervals as in Maillart and Pollock (2002).

### 3.4. Objective

For a process that starts at age \( \tau \), we let \( g(T, \tau|s(\tau), u(\tau)) \) denote the expected total cost incurred during the processing of the next \( T \) products under the \((s, u)\)-policy. We consider a decision maker whose objective is to find the optimal \((s, u)\)-policy (i.e., the optimal action \( s(\tau) \) and \( u(\tau) \) at each possible \( \tau \) value) that mini-
mizes the long-run expected cost per product, i.e.,
\[ \gamma^* = \min_{T \to \infty} \frac{g(T, \tau | s(\tau), u(\tau))}{T}, \]
where the minimization is over all the possible (s, u)-policies, and \( \gamma^* \) is the minimum long-run expected cost per product under the optimal (s, u)-policy. For a process at age \( \tau \), we let \( s(\tau) \) and \( u(\tau) \) denote the values of \( s(\tau) \) and \( u(\tau) \), respectively, under the optimal (s, u)-policy.

The objective function in (1) addresses the trade-off between being less proactive (i.e., higher costs of defective production and corrective maintenance) on one hand and being more proactive (i.e., higher costs of inspection and preventive replacement) on the other hand. Notice that the performance measure is defined as the long-run expected cost per product. This is suitable for a more expensive for an established manufacturer in a competitive market, where the products must be produced as least costly as possible. We emphasize that the duration of inspection and maintenance activities and the time it takes to process one product are not explicitly specified in our model because the objective represents the cost per product, not per time unit.

4. Alert-arrival process

The objective of this section is to characterize the alert-arrival process generated by the prediction model described in Section 3.2. Specifically, we will characterize the probability distribution of the number of products by which a specific number of alerts is generated. We do not explicitly consider the process failures in this section. To be specific, the process starts in the healthy state, it becomes defective after a random amount of products, and it stays defective without observing a process failure. This is an assumption only made in this section for convenience in presentation, and Section 5 continues with the original failure model as introduced in Section 3.1.

The probability distribution of the number of products by which a specific number of alerts is observed will be an input to evaluate the expected cost under the (s, u)-policy and then to determine the optimal actions \( s(\tau) \) and \( u(\tau) \) for a process at age \( \tau \). As illustrated in Fig. 1, inspection is scheduled only for a new process (i.e., at age zero), or for a process which is just inspected and found in the healthy state. Thus, at a decision moment (i.e., the moment at which the next inspection moment is scheduled), the process must certainly be in the healthy state. Therefore, in the remainder of this section, we consider a process which is known to be in healthy state. The number of products processed before observing a specific number of alerts is a random variable that depends on the realization of the duration of healthy state and the imperfection of the defect-prediction model. For a process which is currently in healthy state, we let \( T_i(x) \) denote the random variable that represents the number of products at which the \( i \)th alert (starting from the current age) is generated given that the process completes \( x \) more products in the healthy state before entering the defective state. For example, for a healthy process at a certain age, \( T_2(2) \) denotes the number of products at which the 3rd next alert is observed given that the process becomes defective after completing 2 more products in the healthy state. As another example, \( T_i(0) \) denotes the number of products at which the next alert is observed given that the process has just processed the last product in the healthy state and no more products is processed in the healthy state. Later in Section 5.1, further examples on the realization of \( T_i(x) \) will be illustrated in various scenarios under the (s, u)-policy, see Figs. 4 and 5.

Notice that \( T_i(x) \in \{i, i + 1, \ldots \} \) because the defect-prediction model is run after every product and the \( i \)th alert cannot be observed before processing \( i \) products. We let \( p_{i,x}(\cdot) \) denote the pmf of the random variable \( T_i(x) \). In Proposition 2, we will characterize the function \( p_{i,x}(\cdot) \) for all possible values of \( i \) and \( x \), and this will be an input to evaluate the expected cost under the (s, u)-policy. But first, in Proposition 1, we characterize the pmf of random variable \( T_i(x) \), i.e., the number of products at which an healthy process generates the next alert, given that the process becomes defective after completing \( x \) more products in the healthy state.

Proposition 1. For a process which is known to be in healthy state, the probability that the next alert is generated at the \( t \)th product is given by

\[ p_{i,x}(t) = \begin{cases} \alpha (1 - \alpha)^{-1} & \text{if } t \leq x \\ (1 - \alpha)^{x+1} (1 - \beta) & \text{if } t \geq x + 1 \end{cases} \]

for \( t \in \{1, 2, \ldots \} \) given that \( x \in \{0, 1, \ldots \} \).
For any $x$, it can be verified that the probabilities $p_{1,x}(t)$ associated with each possible value of $t$ sum up to 1; i.e.,

$$\sum_{t=1}^{\infty} p_{1,x}(t) = \sum_{i=1}^{x} p_{1,i}(t) + \sum_{t=x+1}^{\infty} p_{1,x}(t) = \alpha \sum_{i=1}^{x} (1 - \alpha)^{i-1} + (1 - \alpha^x) \beta x^{-1} = 1,$$

because $\sum_{i=1}^{x} (1 - \alpha)^{i-1} = (1 - (1 - \alpha)^x)/\alpha$ and $\sum_{t=x+1}^{\infty} \beta^t x^{-1} = 1/(1 - \beta)$.

Proposition 1 shows that the number of products at which the first alert is generated is likely to be larger as the false positive probability $\alpha$ decreases and the false-negative probability $\beta$ increases. For example, Fig. 2 illustrates $p_{1,i}(\cdot)$ which represents the pmf for the number of products at which the first alert is generated when the process completes 8 more products in the healthy state and becomes defective in the 9th product. Fig. 2 (left) has a higher false-positive probability, making it more likely to observe an alert sooner. In Fig. 2 (both in middle and right), it is unlikely to observe an alert before the 9th product. This is due to low false-positive probability. In Fig. 2 (right), the alert is most likely to be realized as soon as the defect-prediction model is run after the defect arrival (i.e., after the 9th product) because of the lower false-negative probability compared to the case in Fig. 2 (middle).

Notice that the arrival moment of the $i$th alert is equal to the number of products until the $(i-1)$th alert and the number of the $(i-1)$th alert and the $i$th alert. Based on this observation, Proposition 2 builds on the expression characterized in (2) and shows how to recursively calculate the pmf of the number of products at which the process generates the $i$th alert for each given value of $x \in \{0, 1, \ldots\}$.

Proposition 2. For a process which is known to be in healthy state, the probability that the $i$th alert is generated at the $t$th product is given by

$$p_{i,t}(x) = \sum_{z=t-i}^{t-1} p_{t-z}(t) p_{1,max(x-z,0)}(t-z) \quad \text{(3)}$$

for $i \in \{2, 3, \ldots\}$ and $t \in \{i, i+1, \ldots\}$ given that $x \in \{0, 1, \ldots\}$.

5. Optimization of the $(s,u)$-policy

This section presents a stochastic dynamic programming (SDP) formulation to solve for the optimal $(s,u)$-policy by minimizing the long-run expected cost per product. To start with, the possible events that cause a change in the state variables of the dynamic program will be discussed in Section 5.1. In Section 5.2, the SDP formulation will be presented for finite planning horizon of $T$ products with the objective of minimizing the expected total cost. In Section 5.3, the formulation is extended to the case where the objective is to minimize the long-run expected cost per product.

5.1. Possible events under the $(s,u)$-policy

Suppose that the planning horizon consists of $T$ products for a process that is currently at age $\tau$ and known to be in the healthy state (i.e., an inspection at age $\tau$ has revealed that the process is healthy and thus $X \geq \tau$ holds). We represent the state of the system with the pair $(T, \tau)$. Given that the decisions $u(\tau)$ and $s(\tau)$ are taken at age $\tau$, there are three types of events regarding what triggers the next intervention to the process under the $(s,u)$-policy: (1) Process failure, (2) alert-triggered inspection, and (3) planned inspection. The realizations of the random variables $X$, $T_{\text{f}(\tau)}(X - \tau)$ and $H$ determine which event takes place.

Process Failure. The process failure is observed if the following event occurs:

$$\text{Event 1: } [X + H < \tau + \min\{u(\tau), T_{\text{f}(\tau)}(X - \tau)\}] \implies X \geq \tau].$$

In the case of Event 1, an immediate corrective maintenance takes place, and the process renews (i.e., the process age is set to 0) with $T - (X - \tau) - H$ products left in the planning horizon. Consequently, given the realizations $X = x$ and $H = h$, the failure event results in the corrective maintenance cost $C_j$ and the cost of working with a defective process $hC_i$, while the state moves from $(T, \tau)$ to $(T - x + \tau - h, 0)$. A realization of Event 1 is illustrated in Fig. 3 for $X = \tau + 3$ (i.e., the process completed $\tau + 3$ products in the healthy state) and $H = 3$ (i.e., the process completed 3 products in the defective state and failed during the processing of the 4th product). The cost $3C_d + C_f$ is charged as a result of Event 1.

Alert-triggered inspection. An alert-triggered inspection is performed when the $s(\tau)$th alert is observed before a failure or planned inspection. This corresponds to the following event:

$$[\tau + T_{\text{f}(\tau)}(X - \tau) \leq X + H, u(\tau) > T_{\text{f}(\tau)}(X - \tau) \mid X \geq \tau],$$

which is the union of the following two disjoint events:
Event 2A: \[ X \geq \tau + C_{3B}(X - \tau) \quad \text{u}(\tau) > T_{(\tau)}(X - \tau) \] \[ \text{u}(\tau) > T_{(\tau)}(X - \tau) \quad | X \geq \tau \]

Event 2B: \[ X < \tau + C_{3B}(X - \tau) \quad \text{u}(\tau) > T_{(\tau)}(X - \tau) \] \[ \text{u}(\tau) > T_{(\tau)}(X - \tau) \quad | X \geq \tau \]

In the case of Event 2A, the alert-triggered inspection finds the process in the healthy state. Then, no maintenance takes place, only the inspection cost \( I_q \) is charged, and the process age increases to \( \tau + T_{(\tau)}(X - \tau) \) (i.e., it does not renew) with \( T - T_{(\tau)}(X - \tau) \) products left in the planning horizon. Therefore, given the realization \( X = 8 \) and \( T_{(\tau)}(X - \tau) = 8 \), the state moves from \( (T, \tau) \) to \( (T - i, \tau + i) \). A realization of Event 2A is illustrated in Fig. 4 (top) for \( X = 8 \) and \( T_5(7) = 6 \).

In the case of Event 2B, the alert-triggered inspection finds the process in the defective state. Then, an immediate preventive maintenance takes place, and the process renews (i.e., the process age is reset to 0) with \( T - T_{(\tau)}(X - \tau) \) products left in the planning horizon. Consequently, conditional on \( X = 8 \) and \( T_{(\tau)}(X - \tau) = i \), the realization of Event 2B results in the inspection cost \( I_q \), preventive maintenance cost \( C_p \), and the cost of working with a defective process \( (i - x + \tau)C_d \). Also, the state moves from \( (T, \tau) \) to \( (T - i, 0) \). A realization of Event 2B is illustrated in Fig. 4 (bottom) for \( X = 8 \) and \( T_5(3) = 7 \), where the realized cost is \( I_q + C_p + 4C_d \).

**Planned inspection.** A planned inspection is performed when the scheduled inspection moment is reached. This corresponds to the following event:

\[ X + \tau \leq H, \text{u}(\tau) \leq T_{(\tau)}(X - \tau) \quad | X \geq \tau \]

which is the union of the following two disjoint events:

Event 3A: \[ X \geq \tau + C_{3B}(X - \tau) \quad \text{u}(\tau) \leq T_{(\tau)}(X - \tau) \] \[ \quad | X \geq \tau \]

Event 3B: \[ X < \tau + u(\tau) \leq X + H, \text{u}(\tau) \leq T_{(\tau)}(X - \tau) \] \[ \quad | X \geq \tau \]

In the case of Event 3A, the planned inspection finds the process in the healthy state. Then, no maintenance takes place, only the inspection cost \( I_p \) is charged, and the state variables move from \( (T, \tau) \) to \( (T - u(\tau), \tau + u(\tau)) \). A realization of Event 3A is illustrated Fig. 5 (top) for \( u(\tau) = 8 \) when \( X = 10 \) and \( T_5(10) \geq 9 \).

**5.2 Finite-horizon total expected cost**

Let \( g(T, \tau) \) denote the minimum expected total cost for a process that starts as healthy at age \( \tau \) and is operated over a finite planning horizon of \( T \) products by following the optimal \((s, u)\)-policy. The problem of calculating \( g(T, \tau) \) and solving for the optimal \((s, u)\)-policy can be formulated as a stochastic dynamic program with state variables \( (T, \tau) \). First, notice that \( g(0, \tau) = 0 \) for all \( \tau \); i.e., the process is terminated at no cost at the end of the planning horizon regardless of the age of the process. Furthermore, it holds that

\[ g(T, \tau) = \min_{\tilde{s}, \tilde{u}} g(T, \tau | \tilde{s}, \tilde{u}) \] \[ (4) \]

where \( g(T, \tau | \tilde{s}, \tilde{u}) \) is the expected total cost for a healthy process at age \( \tau \) if the actions \( \tilde{s} \) and \( \tilde{u} \) are taken at age \( \tau \) and the optimal \((s, u)\)-policy is followed thereafter. Specifically,

\[ g(T, \tau | \tilde{s}, \tilde{u}) = \sum_{k=0}^{\infty} f_k(X | \tau) \left\{ \sum_{i=0}^{\infty} p_{\tilde{i}X-k}(i) \sum_{h=0}^{T_5(\tau)} f_h(h) \right\} \]

\[ \times \left( C_f + hC_d + g(T - x + \tau - h, 0) \right) \] \[ (5) \]

\[ + \sum_{k=0}^{\infty} f_k(X | \tau) \left\{ \sum_{i=0}^{\min(\tilde{s} - \tilde{u} - 1)} p_{\tilde{i}X-k}(i) \left( I_q + g(T - i, \tau + i) \right) \right\} \] \[ (6) \]

\[ + \sum_{k=0}^{\infty} f_k(X | \tau) \left\{ \sum_{i=\tilde{u}+1}^{\tilde{s} - 1} p_{\tilde{i}X-k}(i) \right\} \times \sum_{h=0}^{T_5(\tau)} f_h(h) \left( I_q + C_p + (i - x + \tau)C_d + g(T - i, 0) \right) \] \[ (7) \]

\[ + \sum_{k=0}^{\infty} f_k(X | \tau) \left\{ \sum_{i=0}^{\tilde{u}} p_{\tilde{i}X-k}(i) \left( I_p + g(T - \tilde{u}, \tau + \tilde{u}) \right) \right\} \] \[ (8) \]
where \( f_x(x|\tau) \) is the pmf of the number of products processed in the healthy state conditional on the fact that the process is healthy at age \( \tau \); i.e., \( f_x(x|\tau) = f_x(x)/P(X \geq \tau) \) for \( x \in \{\tau, \tau + 1, \ldots\} \). The expression in (5) corresponds to observing Event 1 after taking the decision \( \hat{\tau} \) and \( \hat{u} \) at age \( \tau \); i.e., the realized value of \( H \) is strictly smaller than the realized value of \( \tau + \min(\hat{u}, T_\tau(X)) \) so that a failure is observed. The expression in (6) corresponds to Event 2A, where the realized value of \( T_\tau(X - \tau) \) is strictly less than \( \hat{u} \) (i.e., so the alert-triggered inspection occurs before the planned inspection) and less than the realized value of \( X - \tau \) (i.e., so the alert-triggered inspection finds the process in the healthy state). The expression in (7) corresponds to Event 2B, where the realized value of \( T_\tau(X - \tau) \) is still strictly less than \( \hat{u} \) (i.e., so the alert-triggered inspection occurs before the planned inspection) but strictly greater than the realized value of \( X - \tau \) (i.e., so the alert-triggered inspection finds the process in the defective state), with the realized value of \( H \) high enough so no failure is observed. The expression in (8) corresponds to Event 3A, where the realized value of \( T_\tau(X - \tau) \) must be at least \( \hat{u} \) (i.e., so the alert-triggered inspection cannot occur before the planned inspection) and the realized value of \( X - \tau \) must be at least \( \tau + \hat{u} \) (i.e., so the planned inspection finds the process in the healthy state). In contrast, the expression in (9) corresponds to Event 3B, where the realized value of \( X - \tau \) must be strictly less than \( \tau + \hat{u} \) (i.e., so the planned inspection finds the process in the defective state), with the realized value of \( H \) high enough so no failure is observed.

5.3. Long-run expected cost per product under the optimal (s, u)-policy

The objective of this section to characterize the expected cost per product under the optimal (s, u)-policy in an infinite planning horizon (i.e., \( \tau \to \infty \)). We also refer to this value as the long-run minimum average cost rate, and denote it with \( \gamma^* \). Since all cost parameters and the probability distributions for the random variables \( X \) and \( H \) are stationary, it follows that

\[
\lim_{\tau \to \infty} g(T, \tau) = V(\tau) + T \gamma^*
\]

where \( V(\tau) \) is a constant representing the relative cost of beginning the operation with a healthy process of age \( \tau \) (Puterman, 2014). In this limit, (4) becomes

\[
V(\tau) + T \gamma^* = \min_{i=0}^{\infty} \sum_{x=0}^{\infty} f_x(x + \tau|\tau) \left[ \sum_{i=0}^{\infty} p_{x,i}(i) \sum_{h=0}^{\infty} f_h(h) \right]
\]

	\times \left( C_j + hC_d + V(0) + (T-x-h)\gamma^* \right)

\[
+ \sum_{x=0}^{\infty} f_x(x + \tau|\tau) \left[ \sum_{i=0}^{\min(x-1, x)} \sum_{h=0}^{\infty} f_h(h) \right]
\]

\[
+ \sum_{x=0}^{\infty} f_x(x + \tau|\tau) \left[ \sum_{i=0}^{\min(x, x-1)} \sum_{h=0}^{\infty} f_h(h) \right]
\]

\[
+ \sum_{x=0}^{\infty} f_x(x + \tau|\tau) \left[ \sum_{i=0}^{\infty} p_{x,i}(i) \sum_{h=0}^{\infty} f_h(h) \right]
\]

\[
+ \sum_{x=0}^{\infty} f_x(x + \tau|\tau) \left[ \sum_{i=0}^{\infty} p_{x,i}(i) \sum_{h=0}^{\infty} f_h(h) \right]
\]

\[
+ \sum_{x=0}^{\infty} f_x(x + \tau|\tau) \left[ \sum_{i=0}^{\infty} p_{x,i}(i) \sum_{h=0}^{\infty} f_h(h) \right]
\]

for \( \tau \in \mathbb{N}_0 \). Notice that the summations on the right-hand side of Eq. (10) are defined over the entire sample space of the random variables. Therefore, the constant \( T\gamma^* \) can be subtracted from both sides of Eq. (10). The value of constant \( V(0) \) can be arbitrarily set to zero (Puterman, 2014). After reorganizing the terms, Eq. (10) can be equivalently written as follows:

\[
V(\tau) = \min_{i=0}^{\infty} \sum_{x=0}^{\infty} f_x(x + \tau|\tau) \left[ \sum_{i=0}^{\infty} p_{x,i}(i) \sum_{h=0}^{\infty} f_h(h) \right]
\]

for \( \tau \in \mathbb{N}_0 \), where

\[
\begin{align*}
V(\tau | \xi, \tilde{\xi}) &= \sum_{i=0}^{\infty} a_t(i, \tilde{\xi}) V(\tau + i) + b_t(\tilde{\xi}, \xi) V(\tau + \tilde{u}) \\
&\quad - c_t(\tilde{\xi}, \xi) \gamma^* + d_t(\tilde{\xi}, \xi)
\end{align*}
\]

with coefficients

\[
\begin{align*}
a_t(i, \tilde{\xi}) &= \sum_{x=0}^{\infty} f_x(x + \tau|\tau) p_{x,i}(i), \\
b_t(\tilde{\xi}, \xi) &= \sum_{x=0}^{\infty} f_x(x + \tau|\tau) \sum_{i=0}^{\infty} p_{x,i}(i), \\
c_t(\tilde{\xi}, \xi) &= \sum_{x=0}^{\infty} f_x(x + \tau|\tau) \sum_{i=0}^{\infty} p_{x,i}(i) \sum_{h=0}^{\infty} f_h(h) \left[ \min(\tau, x-1) \right] \\
&\quad + \sum_{x=0}^{\infty} f_x(x + \tau|\tau) \left[ \sum_{i=0}^{\infty} p_{x,i}(i) \sum_{h=0}^{\infty} f_h(h) \right] \\
&\quad + \sum_{x=0}^{\infty} f_x(x + \tau|\tau) \left[ \sum_{i=0}^{\infty} p_{x,i}(i) \sum_{h=0}^{\infty} f_h(h) \right] + \tilde{u} b_t(\tilde{\xi}, \xi),
\end{align*}
\]

and

\[
\begin{align*}
d_t(\tilde{\xi}, \tilde{\xi}) &= \sum_{x=0}^{\infty} f_x(x + \tau|\tau) \\
&\quad \times \left[ \sum_{i=0}^{\infty} p_{x,i}(i) \sum_{h=0}^{\min(\tau, x-1)} f_h(h) \right] \\
&\quad + \sum_{x=0}^{\infty} f_x(x + \tau|\tau) \left[ \sum_{i=0}^{\min(x-1, x)} \sum_{h=0}^{\min(\tau, x-1)} f_h(h) \right]
\end{align*}
\]

In Section 6, we discuss our solution approach for the optimization problem in (11) and describe how we solve for the minimum expected cost rate \( \gamma^* \) and the corresponding optimal (s, u)-policy.

6. Solution approach

To start with, the random variable \( X \) is assumed to be geometrically distributed in Section 6.1. This is equivalent to assuming that the process moves from healthy state to defective state with a fixed probability independent of the past. In Section 6.2, this assumption is relaxed, and the solution approach is presented for a general probability distribution for the random variable \( X \).
6.1. Geometrically distributed X

The assumption of geometrically distributed number of products in the healthy state is equivalent to saying that the process moves from healthy state to defective state with a fixed probability independent of the age of the process. Because of this so-called memoryless property, an inspection action that does not lead to preventive replacement also results in the renewal of the system. This is because the remaining length of the healthy state, conditioned on the event that the process is found healthy in an inspection, has the same probability distribution with the length of the healthy state for a new process. Thus, the age of the process at the last inspection is irrelevant and the optimal policy can be summarized by a fixed decision (i.e., a fixed monitoring interval and a fixed number of alerts to wait before triggering an inspection) regardless of the age of the process. In other words, the decision maker always uses the same policy parameters regardless of the age (i.e., \( s^*(\tau) = s^* \) and \( u^*(\tau) = u^* \) for all \( \tau \). As the optimal policy is independent of the age of the process, without loss of generality, we set \( V(\tau) = 0 \) for all \( \tau \in \mathbb{N}_0 \), leading to a special case of the optimization model in (11) given by

\[
0 = \min_{\tilde{s}, \tilde{u}} -c_0(\tilde{s}, \tilde{u}) \cdot \gamma^* + d_0(\tilde{s}, \tilde{u}). \tag{12}
\]

Notice that the functions \( c(\tilde{s}, \tilde{u}) \) and \( d_0(\tilde{s}, \tilde{u}) \) are independent of the process age \( \tau \) for geometrically distributed \( X \). Thus, the value of \( \tau \) is arbitrarily set as zero in (12). The optimal cost rate \( \gamma^* \) can be identified as the optimal objective value of the following linear program:

\[
\begin{align*}
\max_{\gamma} & \gamma \\
\text{s.t.} & -c_0(\tilde{s}, \tilde{u}) \gamma + d_0(\tilde{s}, \tilde{u}) & = V(\tilde{s}, \tilde{u}),
\end{align*}
\]

with \( \gamma \) as an unconstrained decision variable. The optimal number of alerts to trigger an inspection and the optimal monitoring interval under the optimal \((s,u)-\)policy, denoted with \( s^* \) and \( u^* \) respectively, satisfy the equation \( c_0(s^*, u^*) \gamma^* = d_0(s^*, u^*) \).

6.2. Generally distributed X

Notice that the domain of the random variables \( X \) and \( H \) is defined as \( \mathbb{N}_0 \), and therefore, the age of the process can go to infinity in theory, leading to a countably infinite state space for the dynamic program in (11). However, in practice, it is often the case that the process becomes more likely to become defective (if it did not become defective yet) as the process continues. This makes it unlikely to see a process at an age higher than a sufficiently large threshold because the process would be renewed as soon as it is found defective. Accordingly, we truncate the state space of the dynamic program in (11) to eliminate the ages that are unlikely to be reached under any \((s,u)-\)policy. Specifically, let \( N \) be chosen such that \( \mathbb{P}(X > N | X \geq N) < \epsilon \) for a sufficiently small \( \epsilon \). This means that the probability of finding a process, which is known as healthy at age \( N \), again in the healthy state in an inspection at age \( N + 1 \) or later cannot be greater than \( \epsilon \). Consequently, we redefine the truncated state space of the dynamic program in (11) as \( \mathcal{T} = \{0, 1, \ldots, \} \). Notice that the probability distribution of the random variable \( X \) is an input required for the characterization of the expected cost rate under the \((s,u)-\)policy. In Appendix A, we describe how this distribution (together with the distribution of \( H \)) can be estimated from historical data available from previous maintenance cycles. Once the distribution of \( X \) is available, the value of \( N \) can be determined by setting it equal to \( \min\{i: \mathbb{P}(X > i | X \geq i) < \epsilon, i \in \mathbb{N}\} \). In our numerical experiments, we set \( \epsilon = 0.01 \) as it leads to a significantly accurate calculation of the expected cost rate (i.e., the expected cost rate does not change significantly if \( \epsilon \) is decreased even further).

### Table 4

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Mean</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1 Geometric ((p = 1, q = 0.9524))</td>
<td>20</td>
<td>420</td>
</tr>
<tr>
<td>X2 Weibull ((p = 1.5, q = 0.991))</td>
<td>20</td>
<td>200</td>
</tr>
<tr>
<td>X3 Weibull ((p = 2.5, q = 0.9996))</td>
<td>20</td>
<td>79</td>
</tr>
<tr>
<td>H1 Uniform(0,20)</td>
<td>10</td>
<td>37</td>
</tr>
<tr>
<td>H2 Constant</td>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 5

For each \((\alpha, \beta)\) pair, first row is the expected cost rate under the AI policy, and second and third rows are the optimal cost rate \( \gamma^* \) and the values of \( (s^*, u^*) \), respectively, under the optimal \((s,u)-\)policy \((l_0 = 1, l_2 = 2, c_6 = 6, c_7 = 60, c_9 = 1) \) and the distributions X1 and H1.

<table>
<thead>
<tr>
<th>(\beta)</th>
<th>(\alpha)</th>
<th>(\gamma^*)</th>
<th>((s^<em>, u^</em>))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>0.05</td>
<td>1.124</td>
<td>(2.50, 2.50)</td>
</tr>
<tr>
<td>0.64</td>
<td>0.75</td>
<td>0.933</td>
<td>(2.50, 2.50)</td>
</tr>
<tr>
<td>0.65</td>
<td>0.75</td>
<td>0.940</td>
<td>(2.50, 2.50)</td>
</tr>
<tr>
<td>0.65</td>
<td>0.75</td>
<td>0.807</td>
<td>(2.50, 2.50)</td>
</tr>
<tr>
<td>0.67</td>
<td>0.75</td>
<td>0.957</td>
<td>(2.50, 2.50)</td>
</tr>
<tr>
<td>0.67</td>
<td>0.75</td>
<td>0.833</td>
<td>(2.50, 2.50)</td>
</tr>
<tr>
<td>0.67</td>
<td>0.75</td>
<td>0.902</td>
<td>(2.50, 2.50)</td>
</tr>
<tr>
<td>0.69</td>
<td>0.75</td>
<td>0.978</td>
<td>(2.50, 2.50)</td>
</tr>
<tr>
<td>0.69</td>
<td>0.75</td>
<td>0.866</td>
<td>(2.50, 2.50)</td>
</tr>
<tr>
<td>0.72</td>
<td>0.80</td>
<td>1.006</td>
<td>(2.50, 2.50)</td>
</tr>
<tr>
<td>0.72</td>
<td>0.80</td>
<td>0.904</td>
<td>(2.50, 2.50)</td>
</tr>
<tr>
<td>0.72</td>
<td>0.80</td>
<td>0.915</td>
<td>(2.50, 2.50)</td>
</tr>
<tr>
<td>0.72</td>
<td>0.80</td>
<td>0.915</td>
<td>(2.50, 2.50)</td>
</tr>
</tbody>
</table>

The action space is defined as \( A = A(\tau) = \{(s, u) : s, u \in \{1, 2, \ldots, \tau\}, s \leq u \} \) for all \( \tau \in \mathcal{T} \). Notice that it is not necessary to allow \( s > u \) in the action space because this represent the situation where an alert-triggered inspection never takes place, and this scenario is already captured when \( s = u \). The upper limit \( L \) makes the action space finite and can be set to an arbitrarily large value. We solve the dynamic program in (11) with policy iteration, as outlined in Algorithm 1.

In the policy evaluation step of Algorithm 1, notice that the parameter \( V_j \) represent the relative cost of beginning the operation with a healthy process of age \( j \). In the truncated state space with sufficiently large \( N \), we know that the coefficients that are multiplied with \( V_j \) will be zero anyway for \( j > N \) (i.e., the probability of being in those states is negligible under any \((s,u)-\)policy). Thus, without loss of generality, we set \( V_j = 0 \) for \( j > N \). Therefore, the policy evaluation step includes solving a system of linear equations with \( N + 1 \) equations and \( N + 1 \) variables. The policy iteration algorithm terminates when the current policy cannot be improved anymore and the cost rate stays the same. We refer the reader to Bertsekas (2011) for details on the policy iteration algorithm.

7. Numerical experiments

For numerical analysis, we consider that the number of products completed in the healthy state of the production process is represented with a discrete Weibull distribution (Nakagawa & Osaki, 1975; Vila, Nakano, & Saulo, 2019). To be specific, the pmf and cdf of the random variable \( X \) is given by \( f_X(x) = q^{q^k} - q^{(k+1)} \) and \( F_X(x) = 1 - q^{k+1} \) for \( x \in \mathbb{N}_q \), where \( p > 0 \) and \( 0 < q < 1 \) are the shape parameters. When \( p = 1 \), note that this distribution becomes a geometric distribution with success probability \( 1 - q \). The hazard rate is an increasing function for \( p > 1 \), similar to the continuous Weibull distributions. The details of the three discrete Weibull distributions which we use to model the random variable \( X \) are given in Table 42 (see the distributions X1, X2 and X3). For the number
Require:
State space $\mathcal{T}$ and action space $\mathcal{A}$.
Functions $a_t(\cdot, \cdot), b_t(\cdot, \cdot), c_t(\cdot, \cdot)$, and $d_t(\cdot, \cdot)$ for all $\tau \in \mathcal{T}$.
Initialize $\gamma$ and the policy $(s(\tau), u(\tau))$ arbitrarily for all $\tau \in \mathcal{T}$.
Set $\text{converged} = 0$.
while $\text{converged} = 0$ do

Policy evaluation step:
Let $V'_1, \ldots, V'_N$ and $\gamma'$ denote the solution of the system of linear equations:
\[
0 = \sum_{i=1}^{N} a_{i}(i, s(1))V'_i + b_{0}(s(0), u(0))V'_{a(0)} - c_{0}(s(0), u(0))\gamma' + d_{0}(s(0), u(0))
\]
\[
V_i = \sum_{i=1}^{N} a_{i}(i, s(1))V'_{i+1} + b_{1}(s(1), u(1))V'_{a+1} - c_{1}(s(1), u(1))\gamma' + d_{1}(s(1), u(1))
\]
\[
\vdots
\]
\[
V_N = \sum_{i=1}^{N} a_{i}(i, s(N))V'_{N+1} + b_{N}(s(N), u(N))V'_{a+u(N)} - c_{N}(s(N), u(N))\gamma' + d_{N}(s(N), u(N))
\]
with variables $V_1, \ldots, V_N$ and $\gamma$ (i.e., the parameters $V_j = 0$ for $j > N$).

Policy iteration step:
for $\tau \in \mathcal{T}$ do
  for $(\tilde{s}, \tilde{u}) \in \mathcal{A}$ do
    $V(\tau | (\tilde{s}, \tilde{u})) \leftarrow \sum_{\tilde{i}=1}^{N} a_{\tilde{i}}(\tilde{i}, \tilde{s})V'_{\tilde{i}+\tilde{a}} + b_{\tilde{a}}(\tilde{s}, \tilde{u})V'_{\tilde{a}+\tilde{b}} - c_{\tilde{a}}(\tilde{s}, \tilde{u})\gamma' + d_{\tilde{a}}(\tilde{s}, \tilde{u})$
  end for
end for
for $\tau \in \mathcal{T}$ do
  $(\hat{s}(\tau), \hat{u}(\tau)) \leftarrow \text{argmin}_{\tilde{s}, \tilde{u}} \{V(\tau | \tilde{s}, \tilde{u})\}$
end for
Run the policy evaluation step to obtain the cost rate $\gamma'$ under the policy $(\hat{s}(\cdot), \hat{u}(\cdot))$.
if $s(\tau) = \hat{s}(\tau)$ and $u(\tau) = \hat{u}(\tau)$ for all $\tau \in \mathcal{T}$ and $\gamma' = \gamma'$ then $\text{converged} = 1$.
end if
$s(\tau) \leftarrow \hat{s}(\tau)$ and $u(\tau) \leftarrow \hat{u}(\tau)$ for all $\tau \in \mathcal{T}$.
end while
return $s^*(\tau) = \hat{s}(\tau)$ and $u^*(\tau) = \hat{u}(\tau)$ as the optimal action at state $\tau \in \mathcal{T}$, and $\gamma^* = \gamma'$ as the expected cost rate under the optimal policy.

Algorithm 1: Policy iteration algorithm to solve for the optimal $(s,u)$-policy.

of products processed in the defective state, we consider a discrete uniform distribution and a constant value (see the distribution H1 and H2, respectively, in Table 4).

Notice that $X_1$ is the geometric distribution representing a situation where the number of products until becoming defective is independent of the age of the process. The distributions $X_2$ and $X_3$ are selected such that they have the same mean but less variability with an increasing hazard-rate function. For $\epsilon = 0.01$, the parameter $N$ that truncates the state space is equal to 1893 and 279 for distributions $X_2$ and $X_3$, respectively. The distribution $X_1$ captures a situation where a defective process can make any number of products from a given range at equal probabilities. The distribution H2 represents a fixed number of products in the defective state. The so-called deterministic delay time is a reasonable assumption in many practical situations; e.g., when the crack gradually expands at a constant rate in the electro-chemical machining process of PLC mentioned in Section 1. Also, van Oosterom et al. (2014) consider an inspection planning model under deterministic delay time. We study the following inspection-scheduling policies as benchmarks in our numerical analysis:

- **Always inspect (AI)** policy: An inspection is performed if and only if an alert is observed. This is the same as the inspect-at-first-alert policy, mentioned in Section 1. It is a special case of the $(s,u)$-policy with $s(\tau) = 1$ and $u(\tau) = \infty$ for all $\tau$.
- **Alert-Driven Inspection (ADI)** policy: For a process at age $\tau$, an inspection is scheduled to be performed only when the $s(\tau)$-th alert is observed. The ADI policy is a special case of the $(s,u)$-policy with $u(\tau) = \infty$ for all $\tau$, and it includes the AI policy as a special case when $s(\tau) = 1$ for all $\tau$.
- **Age-Based Inspection (ABI)** policy with age-dependent inspection intervals: For a process at age $\tau$, an inspection is scheduled to be performed when the process completes $u(\tau)$ additional products. The availability of the defect-prediction model is ignored. It is a special case of the $(s,u)$-policy with $s(\tau) = u(\tau)$ for all $\tau$ (i.e., recall that an alert-triggered inspection is never possible if this condition holds). This policy is formally analyzed by Maillart and Pollock (2002) in a continuous-time setting.

In the remainder of this section, Section 7.1 quantifies the performance improvement by the optimal $(s,u)$-policy compared to the benchmark policies and discusses when the optimal $(s,u)$-policy becomes the most beneficial. The economic value of an imperfect defect-prediction model under the optimal $(s,u)$-policy is quantified, and it is investigated how this value is affected by the imperfectness of the prediction model. Section 7.2 shows how the structure of the optimal $(s,u)$-policy is affected from various problem parameters.

7.1. Comparison of the optimal $(s,u)$-policy with benchmark policies

The objective of this section is to investigate when it is the most beneficial to generalize the AI and ABI policies to the $(s,u)$-policy. To start with, the number of products processed in the healthy state $X_4$ is assumed to be geometrically distributed (i.e., the distribution $X_1$). For each $(\alpha, \beta)$ pair, Table 5 presents the expected cost rate under the AI policy (first row), the expected cost rate $\gamma^*$ under the optimal $(s,u)$-policy (second row) and the corresponding optimal solution $(s^*, u^*)$ (third row). Notice that the decisions $(s^*, u^*)$ under the optimal policy do not depend on the age
of the process in Table 5 because $X$ is geometrically distributed. We set the upper bound parameter $L$ of the action space equal to 50, which is a sufficiently large value in our context.

There are three types of results presented in Table 5: (1) For sufficiently low values of false-positive probability $\alpha$ (i.e., $\alpha = 0.05$ with $\beta \leq 0.4$, and for $\alpha = 0.1$ and $\beta = 0.4$), we observe that the AI policy turns out to be optimal since it is obtained that $(s^*, u^*) = (1, 50)$ under the optimal policy (i.e., planned inspection is not possible with $u^*$ equal to $L$). Furthermore, for sufficiently accurate defect-prediction model (i.e., $\alpha < 0.2$ and $\beta < 0.3$), the optimal policy turns out to be an ADI policy. (2) When the defect-prediction model is not good enough (i.e., see the entries in italic font for $\alpha = 0.4$ and $\beta \geq 0.2$ and for $\alpha = 0.3$ and $\beta = 0.4$ in Table 5), the optimal $(s, u)$-policy turns out to be the optimal ABI policy (i.e., note that $s^* = u^*$ implies an alert-triggered inspection is never possible). (3) In the remaining cases, we observe that the optimal $(s, u)$-policy achieves a strictly lower expected cost rate than the optimal ADI policy and the optimal ABI policy (i.e., see the entries in bold font in Table 5). These cases can be interpreted as the ones where it is beneficial to generalize both the ABI and ADI (and hence the AI) policies to the $(s, u)$-policy.

For the problem instance used in Table 5, the expected cost rate under the optimal ABI policy is $\gamma_{ABI}^* = 0.915$. We observe that there can be up to 29.2% (i.e., (0.915-0.648)/0.915) reduction in the expected cost rate by generalizing the optimal ABI policy into the $(s, u)$-policy (when $\alpha = 0.05$ and $\beta = 0.05$). When $\alpha = 0.4$ and $\beta = 0.4$, there is 13.4% reduction in the expected cost rate by generalizing the optimal ADI policy into the $(s, u)$-policy, i.e., $(1.056 - 0.915)/1.056 = 13.4\%$, where 1.056 is the expected cost rate of the optimal ADI policy at the corresponding $\alpha$ and $\beta$ values (not shown in Table 5). Let $\Delta$ denote the percentage reduction in the expected cost rate by following the optimal $(s, u)$-policy instead of the optimal ABI policy (i.e., $\Delta = (\gamma_{ABI}^* - \gamma^*)/\gamma_{ABI}^*$). Fig. 6 illustrates the effect of the cost parameters on the percentage benefit $\Delta$. It shows that a process with high corrective maintenance and low alert-driven inspection cost is an ideal candidate to migrate from classical age-based inspection to the proposed hybrid inspection policy. In practice, the decision maker can compare the cost of a prediction model (e.g., the cost of model development or additional sensors for further data collection, etc.) at a certain accuracy level with the economic benefit of moving from optimal ABI policy to the optimal $(s, u)$-policy, and decide whether it is beneficial to add alert-driven inspections to the maintenance program instead of only relying on inspections at predetermined usage intervals. For example, we observe in Fig. 6 (left) that adopting the optimal $(s, u)$-policy brings up to 40% cost reduction by incorporating the alert information on top of a periodic inspection policy.

For the distribution X2 and each $(\alpha, \beta)$ pair, Table 6 presents the expected cost rate under the AI policy (first row) and the expected cost rate $\gamma^*$ under the optimal $(s, u)$-policy (second row). Different from Table 5, Table 6 does not report the optimal actions as the optimal $(s, u)$-policy now depends on the age of the process as well (see Section 72 for illustration of the optimal policy). Table 6 shows that AI policy is still optimal for low false-positive probability (in this case, only when $\alpha = 0.05$) and the optimal ABI policy (with cost rate $\gamma_{ABI}^* = 0.882$) again becomes the optimal $(s, u)$-policy for a defect-prediction model of poor quality (i.e., when $\alpha + \beta \geq 0.7$ in Table 6). In the remaining cases of Table 6, the expected cost rate under the optimal $(s, u)$-policy is always less than the expected cost rate of the AI policy and the optimal ABI policy.

A relevant question is whether (and to what extent) it pays off to invest in further improving prediction accuracy (e.g., via investing in digitalization and analytics expertise) for reducing the expected cost rate. This question can be answered by analyzing how the expected cost rate under the optimal $(s, u)$-policy is influenced by the prediction-quality parameters $\alpha$ and $\beta$. In particular, the optimal cost rate at the current $(\alpha, \beta)$ pair can be compared with the optimal cost rate at a target $(\alpha, \beta)$ pair. If the improvement in the cost rate is higher than the additional cost rate to be incurred due to improving the prediction model, then the decision maker can conclude that it is worth to improve the prediction model. Otherwise, the prediction model can be considered as good enough with no further improvement in $\alpha$ or $\beta$. Intuitively, a prediction model can be tuned to be more sensitive in certain data patterns, making it more likely to trigger an alert leading to a higher false-positive probability $\alpha$. On the other hand, making the prediction-model less sensitive to such patterns leads to a higher false-negative probability $\beta$. Given this trade-off between the false-positive and false-negative probabilities, it is practically relevant to understand how sensitive the cost rate is to a unit change in $\alpha$ or $\beta$. For example,
Table 6 shows that the optimal cost rate is \( \gamma^* = 0.807 \) for \( \alpha = 0.2 \) and \( \beta = 0.2 \). If the false-negative probability \( \beta \) is improved from 0.2 to 0.1, we see the optimal cost rate goes down to \( \gamma^* = 0.779 \). The same amount of decrease in the false-positive probability \( \alpha \), however, is more effective, leading to the reduction of the optimal cost rate down to \( \gamma^* = 0.725 \). In Tables 5 and 6, we always observe that an improvement in the false-positive probability is more beneficial than the same amount of improvement in the false-negative probability.

7.2. The structure of the Optimal (s, u)-policy

In this section, we investigate the properties of the optimal (s, u)-policy. We start with geometrically distributed X, where the optimal (s, u)-policy does not depend on the process age \( \tau \); i.e., \( s^*(\tau) = s^* \) and \( u^*(\tau) = u^* \) for all \( \tau \). In Table 5, we already observed that for a fixed false-positive probability \( \alpha \), as the false-negative probability \( \beta \) decreases, \( u^* \) (i.e., the optimal number of products to process before triggering a planned inspection) is nondecreasing. For example, consider the bold entries in Table 5 with \( \alpha = 0.3 \): we observe that \( u^* \) increases from 4 to 14 as \( \beta \) decreases from 0.3 to 0.05. A higher \( u^* \) value implies that the optimal (s, u)-policy relies more on the number of alerts but less on the number of products processed since last inspection to trigger a new inspection. This is intuitive because it becomes more likely to generate an alert signal for a defective process as \( \beta \) gets smaller. Also, for a fixed \( \beta \), we observe that \( u^* \) is nonincreasing in \( \alpha \), reflecting the fact that the optimal (s, u)-policy relies more on the number of products (rather than the number of alerts) to trigger an inspection when the false-positive probability is high.

Furthermore, we can argue that the optimal (s, u)-policy can be beneficial in its capability of postponing the maintenance action (via the selection of the optimal number of alerts to wait to trigger an inspection) especially when the prediction model is sufficiently accurate and the cost parameter \( C_p \) is low. In Table 7 (with \( C_p = 0 \), \( I_p = I_p = 2 \), \( C_r = 6 \), \( C_f = 30 \) and distribution X1), we investigate how the average value of \( H \) affects the performance of the optimal (s, u)-policy by taking the distribution H2 (with determin-
age of the process. Furthermore, the optimal action \( s^*(\tau) \) is still nonincreasing in process age \( \tau \). However, the monotonicity of \( u^*(\cdot) \) continues to hold only at a fixed value of \( s^*(\cdot) \). For example, we observe in Fig. 8(b) that \( u^*(\tau) \) is nonincreasing in \( \tau \in \{7, \ldots, 12\} \), where \( s^*(\tau) \) is equal to 3. Similarly, Fig. 8(c) shows that \( u^*(\tau) \) is the first nonincreasing in \( \tau \in \{0, \ldots, 8\} \), then it makes a jump as soon as the function \( s^*(\cdot) \) decreases. A jump in the function \( u^*(\cdot) \) is observed only if there is a decrease in the value of the function \( s^*(\cdot) \). But, a decrease in \( s^*(\cdot) \) does not always lead to a jump in \( u^*(\cdot) \).

8. Conclusion

We study the inspection scheduling decisions for a production process that goes through a hidden defective phase which can only be detected via costly inspections. The production process is equipped with a defect-prediction model that generates alert and no-alert signals to indicate the defective and healthy state of the process, respectively. However, the signals generated by the defect-prediction model are not always correct. To address the question of how to react to imperfect alert and no-alert signals, we introduce a new inspection policy, referred to as the \((s, u)\)-policy, to take into account these signals in inspection scheduling. Specifically, the \((s, u)\)-policy schedules the next planned inspection after a specific number of products depending on the age of the process. In addition, it allows an inspection to be triggered by a certain number of alerts also depending on the age of the process. Thus, the proposed policy can be regarded as an extension of classical age-based inspection policy. It is applicable when imperfect information is periodically revealed about a system’s true condition and only costly inspections can identify this true condition.

To find the optimal \((s, u)\)-policy, a stochastic dynamic programming model is formulated with the objective of minimizing the long-run expected cost rate. The performance improvement achieved by the optimal \((s, u)\)-policy is quantified by comparing it to the benchmark policies. The economic value of adding an alert-triggered inspection option to a conventional age-based inspection policy is discussed. Quantifying the performance of the optimal \((s, u)\)-policy at a given level of imperfectness can be used in practice to obtain the economic value of a certain level of improvement in the predictive model. A future research direction is to extend the \((s, u)\)-policy to allow multiple alert types that imperfectly indicate various degradation levels of a system. Another potential research direction is to consider that the false-positive and false-negative probabilities of the defect-prediction model are uncertain and updated by observing the outcomes of the inspections.

Appendix

Proof of Proposition 1. We first consider \( t \leq x \). This represents the case where the alert is realized before the defect. Since an alert is observed for a healthy process with probability \( \alpha \) in each product independently from previous products, we can write

\[
p_{1,x}(t) = \alpha (1 - \alpha)^{t-1} \quad \text{for} \quad t \leq x; \quad \text{i.e., the number of products to see an alert is geometrically distributed with success probability } \alpha.
\]

We next consider \( t > x + 1 \), which represents the cases where no alert has yet arrived by the time the process becomes defective. Since the defect-prediction model gives no alert for a healthy process in each product with probability \( 1 - \alpha \) independently from previous products, the probability that no alert has arrived yet at the \( x \)-th product is given by \( (1 - \alpha)^x \). Given that no alert has arrived by the \( x \)-th product, the first alert arrives at the next \((t - x)\)-th product with probability \( \beta^{t-x-1}(1 - \beta) \). Consequently, the probability of observing the first alert at the \( x \)-th product is given by

\[
p_{1,x}(t) = (1 - \alpha)^x \beta^{t-x-1}(1 - \beta) \quad \text{for} \quad t \geq x + 1. \quad \Box
\]

Proof of Proposition 2. Let \( Y_i(x) \) denote the number of products until the first alert and \( Y_i(x) \) denote the number of products between the \((i-1)\)th and \( i \)th alerts for \( i = 2, 3, \ldots \) given that the process makes \( x \) more products in the healthy state and becomes defective in the \((x+1)\)th product. Note that the number of products to see the \( i \)th alert is equal to the sum of the number of products to see the \((i-1)\)th alert and the number of products to see the first alert after the \((i-1)\)th alert:

\[
T_i(x) = T_{i-1}(x) + Y_i(x)
\]

for \( i \in \{2, 3, \ldots \} \) where \( T_1(x) = Y_1(x) \). Let \( z \) denote the realization of the random variable \( T_{i-1}(x) \). There are two cases two consider:
\( \text{(a) Suppose that } x < z. \text{ This represents the case where the process is already completed its healthy state by the time the } (i - 1) \text{th alert is generated; i.e., the process is already in the defective state in the first product that is processed after seeing the } (i - 1) \text{th alert. Thus, the random variable } Y_i(x) \text{ has the distribution of the random variable } T_i(0). \)

\( \text{(b) Suppose that } x > z. \text{ This represents the case where the process is already in the healthy state by the time the } (i - 1) \text{th alert is generated, and it will become defective after processing } x - z \text{ more products in the healthy state. Thus, the random variable } Y_i(x) \text{ has the probability distribution of the random variable } T_i(x - z). \)

Since \( T_{i-1}(x) \) and \( Y_i(x) \) are independent for a given defect arrival moment \( x \), by using (13), it can be written that

\[
P(T_i(x) = t) = \sum_{z=i-1}^{t-1} P(T_{i-1}(x) = z)P(Y_i(x) = t - z), \quad (14)
\]

where \( P(T_{i-1}(x) = z) = p_{i-1,i}(z) \) and

\[
P(Y_i(x) = t - z) = \begin{cases} p_{1,x-z}(t - z) & \text{if } x > z \\ p_{1,0}(t - z) & \text{if } x \leq z. \end{cases}
\]

The result follows from plugging \( P(T_{i-1}(x) = z) \) and \( P(Y_i(x) = t - z) \) into Eq. (14). □

**Acknowledgment**

The author gratefully acknowledge the support by European Commission H2020-ECSEL Project MANTIS-662189 Cyber Physical System-based Proactive Collaborative Maintenance, and the NWO (Dutch Research Council) project Real-Time Data-Driven Maintenance Logistics.

**Appendix A. Estimation of the distributions of \( X \) and \( H \) from historical data**

Let the probability distribution of \( X \), i.e., the number of products processed in the healthy state of the process, has a known functional form with unknown parameters \( \theta_X \). Likewise, we let the random variable \( H \), i.e., the number of products processed in the defective state of the process, has a known parametric form with unknown parameters \( \theta_H \). The objective of this section is to show how the parameters \( \theta_X \) and \( \theta_H \) can be estimated from an historical data set obtained from the previous maintenance cycles of the process.

Suppose that the historical data includes \( K \) completed maintenance cycles. A maintenance cycle refers to the interval from the moment a new process starts until it goes through a corrective or preventive maintenance. Let \( z_k \) denote the age of the process at the end of maintenance cycle \( k \), and \( y_k \) denote the age of the process at which the last inspection at cycle \( k \) has found the process in the healthy state (i.e., if no inspection was made or no inspection found the process healthy at cycle \( k \), we set \( y_k = 0 \)). There are two cases to consider:

\( \text{(a) The maintenance cycle } k \text{ has ended with a corrective maintenance. This event occurs with probability } P(X + H = z_k, X \geq y_k). \text{ Let } T_k \text{ denote the set of indices of these maintenance cycles in the historical data.} \)

\( \text{(b) The maintenance cycle } k \text{ has ended with a preventive maintenance. This event occurs with probability } P(X + H > z_k, X \geq y_k). \text{ Let } T_k \text{ denote the set of indices of these maintenance cycles in the historical data.} \)

A commonly used method to identify the unknown parameters \( \theta_X \) and \( \theta_H \) is maximum likelihood estimation, where the goal is to identify the values of \( \theta_X \) and \( \theta_H \) that maximize the log-likelihood function of the historical data, given by

\[
\sum_{k \in T_i} \log \sum_{x = y_k} z_k f_X(x; \theta_X) f_H(z_k - x; \theta_H) + \sum_{k \in T_i} \log \sum_{x = y_k} \infty f_X(x; \theta_X) (1 - f_H(z_k - x; \theta_H)) \quad (A.1)
\]

The maximization of the objective function (A.1) can be performed numerically by using standard non-linear optimization packages.

**Appendix B. Additional results on the optimal ADI policy**

In this section, we focus on the ADI policy, which is a special case of the proposed \((s, w, \mu)\)-policy with \( w(\tau) = \infty, \forall \tau \). It is natural to expect that the performance of a purely alert-driven policy depends on how accurate the alerts are. We consider that the number of products processed in the healthy state follows the distribution \( X_1 \). That is, the probability of moving from the healthy state to defective state is independent of the age of the process, so the value of \( s(\tau) \) is fixed under an ADI policy. Let \( s(\tau) = s, \forall \tau \). Fig. B.9 quantifies the performance of the ADI policy as a function of \( s \) at specific values of the false-positive probability \( \alpha \) and false-negative probability \( \beta \).

An immediate observation from Fig. B.9 is that the ADI policy, which is a purely alert-driven inspection policy, is not necessarily better than an age-based inspection policy. It can be seen that there is a compact region of the illustrated \((\alpha, \beta)\) space such that the ADI policy performs better than the optimal ADI policy if \( \alpha \) and \( \beta \) fall into that region; in the remaining part of the \((\alpha, \beta)\) space, the optimal ADI policy is better than the ADI policy. We note that the region where the ADI policy performs better is associated with low \( \alpha \) and \( \beta \) values, representing a high-quality defect-prediction model. Fig. B.9 can provide an important insight to the decision maker: If the aim is to make inspections purely alert driven, it is important to make sure that the false-positive and false-negative probabilities take values lower than certain thresholds. If a prediction model with that quality cannot be achieved, a classical age-based policy works simply better than a purely alert-driven inspection policy.

**Table B.8** and **B.9** consider a specific instance of cost parameters with \( l_a = l_p = 2, C_d = 1, C_F = 6, C_p = 30 \), and reports the cost

<table>
<thead>
<tr>
<th>( \alpha )</th>
<th>( \beta )</th>
<th>( X_1 )</th>
<th>( \Delta X )</th>
<th>( X_2 )</th>
<th>( \Delta X )</th>
<th>( X_3 )</th>
<th>( \Delta X )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>0.05</td>
<td>0.561</td>
<td>36.4</td>
<td>0.547</td>
<td>35.2</td>
<td>0.552</td>
<td>30.7</td>
</tr>
<tr>
<td>0.20</td>
<td>0.582</td>
<td>34.0</td>
<td>0.566</td>
<td>32.9</td>
<td>0.571</td>
<td>28.3</td>
<td></td>
</tr>
<tr>
<td>0.35</td>
<td>0.608</td>
<td>31.0</td>
<td>0.592</td>
<td>29.8</td>
<td>0.597</td>
<td>25.0</td>
<td></td>
</tr>
<tr>
<td>0.50</td>
<td>0.637</td>
<td>27.7</td>
<td>0.630</td>
<td>25.3</td>
<td>0.637</td>
<td>20.0</td>
<td></td>
</tr>
<tr>
<td>0.20</td>
<td>0.05</td>
<td>0.653</td>
<td>25.3</td>
<td>0.636</td>
<td>24.6</td>
<td>0.627</td>
<td>21.3</td>
</tr>
<tr>
<td>0.20</td>
<td>0.683</td>
<td>22.5</td>
<td>0.661</td>
<td>21.6</td>
<td>0.650</td>
<td>18.4</td>
<td></td>
</tr>
<tr>
<td>0.35</td>
<td>0.718</td>
<td>18.5</td>
<td>0.696</td>
<td>17.5</td>
<td>0.682</td>
<td>14.4</td>
<td></td>
</tr>
<tr>
<td>0.50</td>
<td>0.771</td>
<td>12.6</td>
<td>0.741</td>
<td>12.1</td>
<td>0.728</td>
<td>8.6</td>
<td></td>
</tr>
<tr>
<td>0.35</td>
<td>0.05</td>
<td>0.723</td>
<td>18.0</td>
<td>0.695</td>
<td>17.7</td>
<td>0.672</td>
<td>15.6</td>
</tr>
<tr>
<td>0.20</td>
<td>0.752</td>
<td>14.7</td>
<td>0.724</td>
<td>14.1</td>
<td>0.700</td>
<td>12.0</td>
<td></td>
</tr>
<tr>
<td>0.35</td>
<td>0.792</td>
<td>10.1</td>
<td>0.762</td>
<td>9.7</td>
<td>0.737</td>
<td>7.5</td>
<td></td>
</tr>
<tr>
<td>0.50</td>
<td>0.848</td>
<td>3.8</td>
<td>0.817</td>
<td>3.2</td>
<td>0.788</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>0.50</td>
<td>0.05</td>
<td>0.773</td>
<td>12.3</td>
<td>0.741</td>
<td>12.2</td>
<td>0.710</td>
<td>10.8</td>
</tr>
<tr>
<td>0.20</td>
<td>0.806</td>
<td>8.6</td>
<td>0.773</td>
<td>8.3</td>
<td>0.741</td>
<td>7.0</td>
<td></td>
</tr>
<tr>
<td>0.35</td>
<td>0.851</td>
<td>3.5</td>
<td>0.816</td>
<td>3.3</td>
<td>0.781</td>
<td>1.9</td>
<td></td>
</tr>
<tr>
<td>0.50</td>
<td>0.909</td>
<td>-3.1</td>
<td>0.875</td>
<td>-3.7</td>
<td>0.837</td>
<td>-5.1</td>
<td></td>
</tr>
</tbody>
</table>

The benefit of the optimal ADI policy compared to the optimal ABLI policy; the expected cost rate under the optimal ABLI policy is given by \( v_{BL} = 0.882 \) for \( X_1 \), \( v_{BL} = 0.844 \) for \( X_2 \), and \( v_{BL} = 0.796 \) for \( X_3 \).
The expected cost rate under the ADI policy for $I_q = 2$, $I_p = 1$, $C_q = 1$, $C_p = 6$, $C_f = 60$ and distributions X1 and H1. The black plane indicates the expected cost rate under the optimal ABI policy.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>X1 $\gamma_{\text{ABI}}$</th>
<th>Δ%</th>
<th>$\gamma_{\text{ADI}}$</th>
<th>Δ%</th>
<th>$\gamma_{\text{ADI}}$</th>
<th>Δ%</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>0.05</td>
<td>0.486</td>
<td>26.6</td>
<td>0.469</td>
<td>27.5</td>
<td>0.476</td>
<td>24.9</td>
</tr>
<tr>
<td>0.20</td>
<td>0.494</td>
<td>0.510</td>
<td>25.4</td>
<td>0.478</td>
<td>26.2</td>
<td>0.485</td>
<td>23.4</td>
</tr>
<tr>
<td>0.35</td>
<td>0.506</td>
<td>0.511</td>
<td>23.7</td>
<td>0.489</td>
<td>24.5</td>
<td>0.496</td>
<td>21.8</td>
</tr>
<tr>
<td>0.50</td>
<td>0.529</td>
<td>0.511</td>
<td>20.2</td>
<td>0.511</td>
<td>21.1</td>
<td>0.516</td>
<td>18.6</td>
</tr>
<tr>
<td>0.20</td>
<td>0.05</td>
<td>0.538</td>
<td>18.8</td>
<td>0.516</td>
<td>20.3</td>
<td>0.511</td>
<td>19.3</td>
</tr>
<tr>
<td>0.35</td>
<td>0.553</td>
<td>0.530</td>
<td>16.6</td>
<td>0.530</td>
<td>18.2</td>
<td>0.526</td>
<td>17.0</td>
</tr>
<tr>
<td>0.50</td>
<td>0.576</td>
<td>0.554</td>
<td>13.0</td>
<td>0.554</td>
<td>14.4</td>
<td>0.548</td>
<td>13.5</td>
</tr>
<tr>
<td>0.35</td>
<td>0.05</td>
<td>0.572</td>
<td>13.6</td>
<td>0.548</td>
<td>15.4</td>
<td>0.537</td>
<td>15.2</td>
</tr>
<tr>
<td>0.20</td>
<td>0.559</td>
<td>0.571</td>
<td>10.3</td>
<td>0.571</td>
<td>11.8</td>
<td>0.559</td>
<td>11.7</td>
</tr>
<tr>
<td>0.35</td>
<td>0.631</td>
<td>0.607</td>
<td>4.7</td>
<td>0.607</td>
<td>6.3</td>
<td>0.593</td>
<td>6.4</td>
</tr>
<tr>
<td>0.50</td>
<td>0.697</td>
<td>-5.1</td>
<td>-5.1</td>
<td>0.668</td>
<td>-3.2</td>
<td>0.649</td>
<td>-2.4</td>
</tr>
<tr>
<td>0.50</td>
<td>0.05</td>
<td>0.600</td>
<td>9.5</td>
<td>0.577</td>
<td>10.9</td>
<td>0.563</td>
<td>11.0</td>
</tr>
<tr>
<td>0.20</td>
<td>0.630</td>
<td>0.607</td>
<td>4.9</td>
<td>0.607</td>
<td>6.2</td>
<td>0.592</td>
<td>6.6</td>
</tr>
<tr>
<td>0.35</td>
<td>0.679</td>
<td>-2.4</td>
<td>-2.4</td>
<td>0.654</td>
<td>-1.0</td>
<td>0.632</td>
<td>0.1</td>
</tr>
<tr>
<td>0.50</td>
<td>0.755</td>
<td>-13.9</td>
<td>-13.9</td>
<td>0.728</td>
<td>-12.4</td>
<td>0.696</td>
<td>-9.9</td>
</tr>
</tbody>
</table>

Table B.9: The benefit of the optimal ADI policy compared to the optimal ABI policy; the expected cost rate under the optimal ABI policy is given by $\gamma_{\text{ABI}} = 0.663$ for X1, $\gamma_{\text{ABI}} = 0.647$ for X2, and $\gamma_{\text{ABI}} = 0.633$ for X3. The percentage difference of these cost rates is reported, i.e., $\Delta\% = 100(\gamma'_{\text{ABI}} - \gamma'_{\text{ADI}})/\gamma_{\text{ABI}}$. A negative $\Delta\%$ indicates that the optimal ABI policy performs better than the optimal ADI policy.

Fig. B.9. The percentage benefit $\Delta\%$ increases as the defect-prediction model becomes more accurate (i.e., as the parameters $\alpha$ and $\beta$ decrease). At a fixed false-positive probability $\alpha$, it is observed that the speed of increase in the percentage benefit $\Delta\%$ is first high and then decreases as the false-negative probability $\beta$ decreases. In Table B.8 with $\alpha = 0.2$ and the distribution X1, for example, $\Delta\%$ increases 5.9 units as $\beta$ decreases from 0.5 to 0.35, then increases 4 units as $\beta$ decreases from 0.35 to 0.2, and finally increases only 2.8 units as $\beta$ decreases from 0.2 to 0.05. This monotonic change in the speed of increase in $\Delta\%$ holds for all the cases in Tables B.8 and B.9 at a fixed $\alpha$ value. Furthermore, there is a relatively large percentage benefit $\Delta\%$ at low $\alpha$ values. For example, in Table B.8 with $\alpha = 0.05$ and the distribution X1, it is observed that $\Delta\%$ can still be as high as 27.7% even for $\beta = 0.5$, a quite large false-negative probability for a prediction model. In this example with specified cost parameters, the optimal ADI policy, which triggers an inspection only by counting the alerts, becomes more advantageous by improving the false-positive probability rather than improving the false-negative probability. Finally, the values of the percentage benefit $\Delta\%$ are lower in Table B.9 compared to their counterparts in Table B.8. That is, as the number of products to be processed in the defective state becomes more variable, we observe that the optimal ADI policy becomes more beneficial than the optimal ABI policy.
References


