Decision models for preventive maintenance of technical systems

Jussi Leppinen

Aalto University

Aalto University publication series
Doctoral Theses 50/2026

Decision models for preventive maintenance of technical systems

Jussi Leppinen

A doctoral thesis completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, at a public examination held at the lecture hall M1 (Otakaari 1, Espoo, Finland) of the school on 27 February 2026 at 12 noon.

The public defence will also be organized remotely via remote technology.
Link: https://aalto.zoom.us/j/64356888483
Zoom quick guide: https://www.aalto.fi/en/services/zoom-quick-guide

Aalto University
School of Science
Department of Mathematics and System Analysis

Supervising professor
Professor Ahti Salo, Aalto University, Finland

Thesis advisors
Professor Ahti Salo, Aalto University, Finland
D.Sc. (Tech.) Antti Punkka, S Group, Finland

Preliminary examiners
Professor Jutta Geldermann, University of Duisburg-Essen, Germany
Associate Professor Bram de Jonge, University of Groningen, Netherlands

Opponent
Professor Jutta Geldermann, University of Duisburg-Essen, Germany

Aalto University publication series
Doctoral Theses 50/2026

ISBN 978-952-64-3008-9 (paperback)
ISBN 978-952-64-3007-2 (pdf)
ISSN 1799-4934 (print)
ISSN 1799-4942 (pdf)

http://urn.fi/URN:ISBN:978-952-64-3007-2

PunaMusta Oy
2026

Aalto University

ABSTRACT

Author Jussi Leppinen

Name of the doctoral thesis Decision models for preventive maintenance of technical systems

Article-based thesis

Number of pages 113

Keywords maintenance scheduling, multi-component system, Markov decision process, downtime, stage-gate process

The reliable operation of technical systems and their components requires maintenance, the costs of which can constitute a substantial share of a system's total operating expenses. Rather than relying on corrective maintenance, systems are often best serviced preventively with the aim of eliminating failures, since downtime can be expensive. However, components should not be replaced more frequently than necessary. Consequently, optimizing component maintenance schedules can yield significant cost savings. Yet, optimization is complicated by dependencies between components and by incomplete information about their remaining useful life.

This dissertation presents novel analytical approaches to improve the maintenance management of technical systems. It has two main contributions. The first contribution is a Markov decision process model to determine cost-effective maintenance schedules for systems when there are economic and structural dependencies between components. The second contribution is a structured decision-making process that supports the development of AI-based solutions for predicting the remaining useful life of the system.

The proposed Markov model is the first to account for the threshold for system reliability and the structural dependencies arising from the disassembly and replacement of components. In addition, it considers reliability structures derived from various component configurations and the expected costs of system downtime when maintenance can be performed periodically. The optimal maintenance policy is solved using a modified policy iteration algorithm enhanced with Anderson acceleration. The resulting maintenance schedule leverages the simultaneous servicing of components, achieving improved cost efficiency in comparison with many heuristic policies. The model supports operational, tactical, and strategic maintenance decision-making by providing insights into the use of individual maintenance actions, long-term total costs of the optimal policy, and, for example, the impact of the maintenance period length on overall costs. Furthermore, the model offers guidance on how to design the system structure.

No process has previously been established for developing AI-based solutions aimed at predicting a system's remaining useful life while taking into account both technical and organizational criteria in the early stages of development. In the proposed stage-gate process, the development of AI-based solutions progresses from use case definition through exploratory testing and implementation planning to final deployment. At each decision gate, the candidate solutions to advance are selected. Candidates are evaluated against both technical and organization-specific criteria, which are derived from six development objectives. Since uncertainties may exist in the evaluation of candidates with respect to criteria and prioritization of objectives during early development stages, robust decision-making is employed to support choices at the gates. This reduces the risk that selected candidates later become infeasible. In addition, the benefits of the development effort are assessed in terms of the organization's capability advancement.

Aalto-yliopisto

TIIVISTELMÄ

Tekijä Jussi Leppinen

Väitöskirjan nimi Päätösmalleja teknisten järjestelmien ennakoivaan kunnossapitoon

Artikkeliväitöskirja

Sivumäärä 113

Avainsanat kunnossapidon aikataulutus, monikomponenttijärjestelmä, Markovin päätösprosessi, seisokkiaika, vaiheistettu päätösprosessi

Teknisten järjestelmien ja järjestelmien komponenttien luotettava toiminta vaatii kunnossapitoa, jonka kustannukset saattavat olla huomattava osa järjestelmän käyttökustannuksista. Korjaavan kunnossapidon sijaan järjestelmää on usein syytä huoltaa ennaltaehkäisevästi rikkoutumisten estämiseksi, koska järjestelmän toimimattomuus voi aiheuttaa merkittäviä kustannuksia. Komponentteja ei kuitenkaan tulisi vaihtaa tarpeettoman usein. Täten komponenttien kunnossapitoaikataulujen optimoinnilla voidaan usein saavuttaa merkittäviä kustannussäästöjä. Haasteena optimoinnissa ovat komponenttien väliset erilaiset riippuvuudet sekä puutteellinen tieto komponenttien jäljellä olevasta käyttöiästä.

Tässä väitöskirjassa kehitetään uusia analyyttisia lähestymistapoja teknisten järjestelmien kunnossapidon tueksi. Sillä on kaksi pääkontribuutiota. Ensimmäinen kontribuutio on Markovin prosessiin perustuva päätösmalli, jolla määritellään järjestelmän komponenteille optimaalinen kunnossapidon aikataulu, kun komponenttien välillä on taloudellisia ja rakenteellisia riippuvuuksia. Toinen kontribuutio on vaiheistettu päätösprosessi, jolla voidaan ohjata tekoälypohjaisten ratkaisujen kehittämistä tehtäviin, joissa on arvioitava järjestelmän jäljellä oleva käyttöikä.

Markov-malli on laatuaan ensimmäinen, joka huomioi järjestelmän luotettavuuden rajoittamisen sekä komponenttien purkamisesta ja uusimisesta aiheutuvat rakenteelliset riippuvuudet. Lisäksi tämä huomioi eri kokoonpanoista johtuvat luotettavuusrakenteet ja järjestelmän toimimattomuuden odotusarvoiset kustannukset, kun kunnossapitoa tehdään vain vakiopituisten periodien välein. Optimaalinen huoltopolitiikka ratkaistaan muokatulla politiikan iterointi -algoritmilla, jota tehostetaan Andersonin kiihdytyksellä. Optimoitu kunnossapidon aikataulu hyödyntää komponenttien samanaikaista huoltamista ja on kustannustehokkaampi kuin monet heuristiset huoltopolitiikat. Malli tukee kunnossapidon operatiivista, taktista ja strategista päätöksentekoa tarjoamalla tietoa yksittäisistä huoltopäätöksistä, pitkän aikavälin kokonaiskustannuksista ja esimerkiksi huoltojakson pituuden vaikutuksesta kokonaiskustannuksiin. Lisäksi malli antaa tukea sille, miten järjestelmän rakenne tulisi suunnitella.

Järjestelmän jäljellä olevan käyttöiän ennustamiseen tähtäävien tekoälyratkaisujen kehittämisen tueksi ei ole aikaisemmin laadittu prosessia, joka kehitystyön alkuvaiheessa huomioisi sekä tekniset että organisaatiokohtaiset kriteerit. Vaiheistetussa päätösprosessissa ratkaisujen kehittäminen etenee käyttötapausten määrittelyn sekä tutkivan testauksen ja toteutussuunnitelman kautta ratkaisun varsinaiseen toteutukseen. Vaiheiden välisissä päätösporteissa valitaan ratkaisuehdokkaat, joiden kehitystyötä jatketaan. Ehdokkaita arvioidaan sekä teknisillä että organisaatiokohtaisilla kriteereillä, jotka pohjautuvat kuuteen kehitystavoitteeseen. Kehitystyön alkuvaiheissa voi olla epävarmuuksia sekä ehdokkaiden arvioinneissa eri kriteerien suhteen että kehitystavoitteiden priorisoinnissa. Päätösporteissa tehtävissä valinnoissa hyödynnetään robustia päätöksentekoa, mikä vähentää riskiä sille, että kehitystyön alla olevat ehdokkaat osoittautuisivat myöhemmin sopimattomiksi. Kehitystyön hyötyjä arvioidaan lisäksi organisaation kyvykkyyksien kehittymisen näkökulmasta.

Preface

Pursuing a doctoral degree was not my goal when I started my university studies. I was not even familiar with the term operations research when I began my mathematics studies at the University of Helsinki in the fall of 2013. After all these 12 years, I see the path to this point as a series of successive coincidences. One of my most spontaneous decisions was to apply to another master's program at Aalto University. At the beginning of those studies, I had an inspiring conversation with Professor Harri Ehtamo, which motivated me to choose operations research as my major.

However, the path towards a doctoral dissertation really opened up when I got a position to write my master's thesis under the supervision of Professor Antti Punkka at the Systems Analysis Laboratory in the summer of 2019. On top of that, there was an EU project whose goals guided two of my articles. Thus, Antti started as my doctoral thesis supervisor. He also introduced the Finnish Operations Research Society to me. Over the past three years, I have worked as a secretary of the society. In this role, I have met many new people from the field of operations research, helped organize various events, and collaborated with other board members of the society. I want to thank Antti for his support and for enabling many of these endeavours.

Later in my doctoral studies, I moved under the supervision of Professor Ahti Salo. Ahti's academic experience has been very valuable to me. His feedback has helped me improve my academic writing. In addition, the discussions with him, related to my articles and research, helped me broaden my views on many things. I want to thank Ahti for all the supervision and collaboration.

I also enjoyed working with undergraduate students, both as a course assistant for Decision Analysis and as a bachelor's thesis advisor. Many of those bachelor students worked in the lab during the summer, which allowed for a more active interaction and more significant results that benefited my research. I have always found the role of a mathematics teacher enjoyable. Positive feedback from students has helped maintain high morale.

Overall, the five years of doctoral studies passed quickly and were a valuable experience. I have learned a lot about working with a persistent mindset, writing in a concise, straightforward style, coding efficiently, presenting results, and speaking in public. Although it has been independent work for most of the time, I have enjoyed the social interactions with students, peers, and professors. I hope that the challenges of the future are equally rewarding.

Finally, I want to thank my parents, siblings, friends, and partner for always being there when I needed support. They also helped me balance my academic work with enjoyable, non-academic free time.

Espoo, February 2, 2026,

Jussi Leppinen

Contents

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I Jussi Leppinen, Antti Punkka, Tommi Ekholm, Ahti Salo. An optimization model for determining cost-efficient maintenance policies for multi-component systems with economic and structural dependencies. *Omega*, 130, 103162, January 2025.

II Jussi Leppinen, Ahti Salo. Balancing downtime and maintenance costs for multi-component systems with economic and structural dependencies. Submitted to *Reliability Engineering & System Safety*, October 2025.

III Jussi Leppinen, Ahti Salo, Michele Compare. A stage-gate decision process for guiding the development of AI solutions for preventive maintenance. *EURO Journal on Decision Processes*, 14, 100063, 2026.

Author’s Contribution

Publication I: “An optimization model for determining cost-efficient maintenance policies for multi-component systems with economic and structural dependencies”

Leppinen is the primary author. He built the directed graph framework for multiperiodic maintenance under the guidance of Punkka and Ekholm. Leppinen implemented this framework as a decision model, identified and expanded solution algorithms for solving this model, and also derived and evaluated computational results. Salo contributed to the positioning and exposition of the paper.

Publication II: “Balancing downtime and maintenance costs for multi-component systems with economic and structural dependencies”

Leppinen is the primary author. He initiated the topic, developed the mathematical formulations, implemented the decision model, and obtained and analyzed numerical results. Salo suggested the idea of accounting for downtime costs with binary decision diagrams. He also contributed to the finalization of the manuscript.

Publication III: “A stage-gate decision process for guiding the development of AI solutions for preventive maintenance”

Leppinen is the primary author. He developed the process together with Salo and Compare. Leppinen carried out the literature review, took part in designing the case study, provided most evaluation statements, and calculated the numerical results. Salo contributed to the definition of objectives and criteria and the finalization of the paper. Compare wrote the taxonomy of AI approaches and suggested the illustrative example presented in the case study.

List of Figures

List of Tables

Abbreviations

AI Artificial intelligence

CBM Condition-based maintenance

CDF Cumulative distribution function

MCS Multi-component system

MDP Markov decision process

PHM Prognostics and health management

1. Introduction

1.1 Maintenance management

A technical system does not operate as intended with complete certainty throughout its life cycle. It consists of many components that may fail, and even the failure of a single component can lead to system failure. Fortunately, system failures are often temporary, allowing the system to be returned to its intended operational condition through *corrective* maintenance on the failed component(s). However, this can be expensive, as the failed system may cause additional expenses or give rise to lost revenues due to its inability to perform its function from the time of failure until corrective maintenance is completed. This time interval is referred to as the system's *downtime*.

Yet, corrective maintenance is not always realistic. Some systems are safety-critical, and their failures should be avoided at all costs. Typically, components degrade during use, which increases their likelihood of failure and decreases their *reliability* to function as intended. Therefore, the chance of component failures can be reduced through *preventive* maintenance. From the viewpoint of minimizing maintenance costs, a component should undergo preventive maintenance just before it fails. However, it is challenging to predict the precise moment of failure. Here, a rise in corrective maintenance expenses occurs if preventive maintenance is delayed too much for short-term savings. For example, if preventive maintenance for a car is ignored, the risk of the car breaking down during operation rises. Some failures, such as engine timing belt breakage, can result in an expensive engine overhaul.

In summary, *maintenance* aims to (i) control or prevent a failure of an engineered system and (ii) restore the system to an operational state after a failure (Ben-Daya et al., 2016). *Maintenance management* oversees all aspects of maintenance, including planning, execution, monitoring, budgeting, workforce management, and continuous improvement to ensure that maintenance is aligned with the strategic goals set for the system operation (Garg and Deshmukh, 2006). These different aspects of maintenance management may be unfamiliar, for example, to a typical car owner who purchases car maintenance as a service that is carried out regularly.

In more complex systems, like aircraft and power generation systems, maintenance management can account for a significant portion of the system's operational costs (Bevilacqua and Braglia, 2000). Thus, optimal scheduling of preventive and corrective maintenance actions can yield substantial cost savings, for example, by minimizing downtime (e.g. Levitin et al., 2023). Typically, the optimal maintenance schedule is one that balances less expensive preventive maintenance with more expensive corrective maintenance, as preventive maintenance does not guarantee that the system will operate without failures.

The role of maintenance has evolved over time (Sherwin, 2000). Early technical systems were maintained by the craftsmen who manufactured them. Repairs were integrated with the evolution of system designs. If a component in the system failed, it was assumed to be too weak and was replaced with a more durable one in the following designs. After the manufacturing of system components was harmonized, spare components became available. This shifted the maintainer's work from craft skills towards the diagnostic ability to detect, isolate, and identify failures when they occur (Jardine et al., 2006). The first scientific approaches to maintenance management date from the 1950s, when preventive maintenance was advocated as a means to reduce failures and unplanned downtime (Dekker, 1996).

Maintenance has long been viewed as a necessary step after failures, but also challenging to manage (de Jonge and Scarf, 2020). Due to the rapid evolution of IT technologies and data collection, maintenance management has become more diversified. Maintenance is no longer just about spending money, but a function that can contribute to profits (Sherwin, 2000). Against this backdrop, this dissertation connects the use of maintenance scheduling decision models with operational, tactical, and strategic planning.

1.2 Maintenance of multi-component systems

In this dissertation, a *system* refers to a multi-component system (MCS) whose components require maintenance to maintain the system's operational status. There can be economic, structural, stochastic, and resource dependencies between components of the MCS (Olde Keizer et al., 2017). The focus of this dissertation is primarily on economic and structural dependencies.

Economic dependence is positive if the simultaneous maintenance of components is less expensive than maintaining them separately. Examples of this include the cost of transporting the system to a location where maintenance actions can be performed, as well as a mechanic's travel expenses. Conversely, negative economic dependence makes it more expensive to maintain components simultaneously. It can occur when the maintenance of multiple components causes extra downtime periods or associated costs. For example, a delay in the car's service can be more expensive, as the car owner is unable to travel as effectively.

According to Nicolai and Dekker (2008), *structural dependence* means that the maintenance of specific components requires the replacement or disassembly of some other components. For example, a gearbox must be separated from the engine to replace a clutch. Later, structural dependence has been analysed further by differentiating between technical and performance dependence (Olde Keizer et al., 2017). *Technical dependence* means that other actions restrict specific maintenance actions. Examples of related studies focus on disassembly sequences (Zhou et al., 2015) and the impact of disassembly on degradation (Dinh et al., 2020). However, the research on technical dependencies remains limited (de Jonge and Scarf, 2020).

Performance dependencies depend on the configuration of the components of the system. They explain, for example, how the reliability of the system is connected to the reliability of its components. According to Olde Keizer et al. (2017), when multiple types of component dependencies are included in the study, structural dependence is usually modelled as a performance dependence, for instance, so that the system has a series, parallel, series-parallel, or k-out-of-N reliability structure between the components of the system. Overall, one of the aims of this dissertation is to develop a new modeling framework that accounts for both technical and performance dependencies.

The timing of preventive maintenance for single-component systems is well understood based on reliability theory (Barlow and Proschan, 1996). The maintenance scheduling of MCSs is more challenging than that of single-component systems. When economic or structural dependencies are present, it is often cost-efficient to maintain several components simultaneously rather than maintaining each component according to its optimal schedule (Olde Keizer et al., 2017). In particular, if a single component fails and corrective maintenance must be done, it is often cost-efficient to conduct preventive maintenance on other components of the system at the same time. This *opportunistic maintenance* has been explored in the maintenance literature (Ab-Samat and Kamaruddin, 2014). This dissertation supports the grouping of maintenance actions and demonstrates that sometimes the optimal grouping can be counterintuitive. Such findings highlight the relevance and importance of using decision models for scheduling maintenance actions.

1.3 Prognostics and health management

Various strategies have been developed to schedule preventive maintenance actions. In the late 1970s, maintenance trends shifted toward condition-based maintenance (CBM) due to advances in sensor technology (Ben-Daya et al., 2016). CBM is a maintenance program that recommends maintenance decisions based on the information collected through condition monitoring (Jardine et al., 2006). CBM started as a policy, where preventive maintenance is triggered when monitoring reveals that the component has degraded below a given threshold. However, from the economic perspective, it is not realistic to monitor all components of the MCS, as each component requires its specific instrument (Bouvard et al., 2011).

Later, CBM has evolved from a reactive to a proactive policy with the help of *prognostics*, which addresses fault prediction by assessing the likelihood of a fault and when it is likely to occur (Jardine et al., 2006). These predictive concerns are central to the emerging discipline of *prognostics and health management* (PHM). PHM makes it possible to view the overall health state of systems and assists in making informed decisions regarding their maintenance (Atamuradov et al., 2017). PHM is employed to enhance system availability and safety. Its applications are not limited to maintenance. A wearable smartwatch is a concrete example of a PHM application; it can determine the user's sleep quality (diagnostics) and alert to the possibility of an upcoming flu based on heart rate variability (prognostics).

Prognostic methods can be categorized into statistical and artificial intelligence-based (AI-based) data-driven methods, physics-based methods, and hybrid methods that combine data-driven and physics-based methods (An et al., 2015). AI-based data-driven methods have received increasing attention due to the ability of modern technologies to collect and analyze data (Lei et al., 2018). However, developing PHM models is difficult because each model must be built on a case-by-case basis, and each case can have different performance requirements and limiting factors. For example, constructing a physics-based method can be challenging if the behavior of the degrading system under dynamic operation conditions is poorly understood. Thus, data-driven models may be needed; however, these can suffer from a lack of run-to-failure data or sensor noise, resulting in suboptimal models with an open interpretation of why the requirements have not been met (Khan et al., 2021). Furthermore, the data that is available for data-driven models may not be accessible to academics for model development, as companies view it as a business benefit not shared with outsiders (Lei et al., 2018).

Even in cases where data is available and a prognostics task is defined, developing an AI-based method to support PHM is not straightforward. Developing a method that meets the performance requirements with minimal expense and within a short timeframe can be advantageous. However, potential methods may be overlooked if the primary development objectives are solely the minimization of time and cost. This motivates the consideration of broader evaluation frameworks to guide the development of AI-based methods.

1.4 Research objectives

This dissertation contributes to two interrelated areas of research. First, models to schedule maintenance actions for MCSs with economic, technical, and performance dependencies are formulated. Second, a decision process for guiding the development of AI-based methods that support PHM is introduced. These contributions are synergistic, for example, allowing the models for MCSs to be used to evaluate the economic impacts of the developed AI-based methods.

Paper I constructs a Markov decision process (MDP) model (Puterman, 2014) to determine optimal maintenance policies that meet reliability requirements for system operation in a periodic setting where there can be economic and technical dependencies between components. The dependencies are modelled with a directed graph. The MDP model is solved with a modified policy iteration that employs Anderson acceleration (Geist and Scherrer, 2018). The optimization model supports decision-making at many levels, including operational, tactical, and strategic levels.

Paper II expands the MDP model of Paper I in two directions. First, the MCS can have any binary reliability structure where the system is either operational or failed, depending on which of its components are operational or failed. This makes the model of Paper I more relevant to many real-world problems. Second, the expected downtime costs are modelled by deriving the expected system downtime during every state transition. The reliability requirement, economic, and technical dependencies of Paper I are included. The model can be used to compare the design alternatives of MCSs in terms of expected maintenance costs and downtime. Additionally, computational aspects of efficiently solving the optimal policy are analyzed, also focusing on how the runtime scales with the problem size.

Paper III introduces a stage-gate process (Cooper, 1990) to guide the development of an AI-based method to support PHM. The development process is divided into a sequence of stages. After each stage, decisions concerning the continuation and termination of method candidates are made at the next decision gate. This process helps accumulate evidence on candidates transparently, control resource allocation, and reduce the risk of prematurely terminating the development of potential candidates while accounting for the strategic goals of the organization responsible for the development work.

2. Methodological background

This chapter introduces the core methodological background of this dissertation. First, Section 2.1 presents the Markov decision process model and standard solution algorithms for it. Section 2.2 specifies relevant concepts from reliability theory. Section 2.3 outlines common assumptions of maintenance scheduling of multi-component systems and introduces directed graphs and binary decision diagrams for modelling structural dependencies. Finally, Section 2.4 introduces the stage-gate process.

2.1 Markov decision processes

Markov decision processes (MDPs) are models for sequential decision-making (Puterman, 2014). They became widely cited in the operations research literature in the 1950s through the work of Bellman and Howard (see, e.g., Howard, 1960). In MDP, the state of a system represents its condition. The states are observed at decision instances. As the system's condition evolves, its state typically changes from one decision instance to the next through an uncertain state transition. The probability of transitioning to a state can be impacted by decisions that, depending on the modeling perspective, accumulate rewards or costs. The goal is to identify actions in each state such that these actions either maximize rewards or minimize costs in the long run. MDPs have limitations. For example, they only support periodic state transitions and decision instances and cannot handle uncertainties in observations (Zhao et al., 2025). Semi-MDPs (Chen and Trivedi, 2005) and Partially Observable MDPs (Corotis et al., 2005) can circumvent these limitations, but they are not applied in this dissertation.

In this dissertation, the MDP models consist of decision instances, states, actions, costs, and transition probabilities. Decision instances t^k are discrete with $k \in \mathbb{N}$. A finite state space S defines the possible states $s \in S$. For each state, there is a set of feasible actions A_s from which action $a \in A_s$ is selected. The action space is $A = \cup_{s \in S} A_s$. The state and action spaces do not depend on decision instances. Choosing action a in state s generates a cost $c(s,a)$. In the next decision instance t^{k+1}, the system attains state s' with probability $p(s'|s,a)$, based on a conditional probability distribution that depends on the state and action on the current instance t^k. The property $\sum_{s'\in S} p(s'|s,a) = 1$ holds for all $s \in S$ and $a \in A_s$.

For systems whose performance is measured by a binary indicator, without distinguishing between different operational levels, the system's operational status in any state $s \in S$ can be modelled simply as either *operational* or *failed*. Thus, the state space can be partitioned into $S = S^O \cup S^F$ where S^O denotes the states in which the system is operational and S^F the states in which the system is failed. Now, a reliability requirement can be imposed to ensure that after each feasible action, the system transitions to a state in which it is operational with a probability higher than threshold $\rho \in (0,1)$: $\sum_{s'\in S^O} p(s'|s,a) \geq \rho$ for all $s \in S$ and $a \in A_s$.

A solution for an MDP is a policy U that assigns the optimal action to each state: $U(s) = a \in A_s$. A *stationary* policy only depends on the state and not on the decision instance t^k. In the rest of this section, we consider solving the discounted MDP to determine the optimal policy that minimizes the discounted total costs cumulated over an infinite time horizon.

Theoretically, the optimal policy can be found by brute-force testing the discounted total costs of every possible policy (at most $|S|^{|A|}$). Yet, this is often computationally infeasible because practical MDP models have large state and action spaces. Thus, the optimal policy is typically determined iteratively.

Specifically, the optimal policy for discrete-time MDPs is determined through a value vector $v^U \in \mathbb{R}^{|S|}$ which is a solution to the Bellman equation

$$v^U(s) = c(s, U(s)) + \beta \sum_{s'\in S} p(s'|s, U(s)) v^U(s'), \tag{2.1}$$

where $0 \leq \beta < 1$ is the discount factor. If $\beta = 0$, only the cost of the current state matters. The component of the value vector $v^U(s)$ represents the expected discounted total costs over an infinite horizon when the system is currently in state s and policy U determines the corresponding action at each decision instance.

The optimal value vector can be computed with standard *value* and *policy iteration* algorithms (Howard, 1960). *Value iteration* starts from an initial value vector guess v^0 and iteratively updates the components of the vector as

$$v^{n+1}(s) = \min_{a\in A_s} \left\{ c(s,a) + \beta \sum_{s'\in S} p(s'|s,a) v^n(s') \right\}. \tag{2.2}$$

This sequence of value vectors converges in norm to the optimal value vector (Puterman, 2014). After convergence, the actions that minimize the components of the value vector in the final iteration are assigned to the optimal policy. Value iteration is conceptually simple and easy to implement, but convergence can take many iterations. The efficiency of value iteration can be increased with splitting methods, such as the Gauss-Seidel method (Puterman, 2014), in which the components of the value vector are updated sequentially, and the updates replace the old values as soon as they become available.

In contrast, the *policy iteration* starts from an initial policy guess U^0. The policy is updated iteratively until the algorithm converges to the optimal policy. The values for the policy are solved from the system of linear equations of the Bellman equation (2.1), which can be written in matrix form as

$$v = c_{U^n} + \beta P_{U^n} v \Leftrightarrow v = (I - \beta P_{U^n})^{-1} c_{U^n}. \tag{2.3}$$

In (2.3), the current policy U^n determines the actions that define costs c_{U^n} and transition probabilities P_{U^n}. Matrix $I - \beta P_{U^n}$ is invertible because βP_{U^n} has spectral radius less than one (Puterman, 2014). After solving the value vector v, the policy update for any state s is

$$U^{n+1}(s) = \arg\min_{a \in A_s} \left\{ c(s,a) + \beta \sum_{s' \in S} p(s'|s,a) v(s') \right\}. \tag{2.4}$$

Policy iteration converges when $U^{n+1}(s) = U^n(s)$ for all $s \in S$. Compared to value iteration, policy iteration converges with fewer iterations, but calls for the computationally demanding operation of solving a system of linear equations of size $|S|$ in each iteration. In discounted MDPs, both value and policy iteration algorithms are guaranteed to converge on condition that the state and action spaces are finite, costs are bounded, and parameters for costs and transition probabilities do not change over time (Puterman, 2014).

The modified policy iteration algorithm (Puterman, 2014) combines the two algorithms mentioned above. It also starts by assuming an initial value vector. Then, the algorithm alternates between a policy update and a fixed number of value computations. The policy is updated according to (2.4). Value computations consist of the value updates (2.2), but instead of determining the action based on solving a minimization problem, the action is determined by the policy. The policy updates and value computations are carried out until the value vector converges. Convergence is tested after each policy update, and the optimal policy is defined by the most recent update.

Typically, the modified policy iteration outperforms both value and policy iterations by requiring fewer iterations for convergence and less computational resources too (see e.g., Andersen et al., 2022). The performance of the modified policy iteration can be further improved with the Gauss-Seidel splitting method and Anderson acceleration (Geist and Scherrer, 2018). Anderson acceleration speeds up the convergence of the value vector by computing a new estimate for the value vector from a weighted sum of previous estimates obtained during the value computation step. Weights for the previous estimates are derived through optimization.

2.2 Reliability theory

According to Barlow and Proschan (1996), mathematical reliability theory is “a body of ideas, mathematical models, and methods directed toward the solution of problems in predicting, estimating, or optimizing the probability of survival, mean life, or, more generally, life distribution of components or systems.” Let $\mathrm{P}(\cdot)$ denote probability measure over domain $[0, \infty)$. A nonnegative random variable X, such as a failure age of a component, can be modelled using a cumulative distribution function (CDF) $F(\cdot)$ such that $\mathrm{P}(X \leq x) = F(x)$. The expected value of X is

$$\mathrm{E}[X] = \int_0^\infty (1 - F(x))\, dx. \tag{2.5}$$

Moreover, if the component has not failed before its age reaches a_1, it will not fail before age $a_2 > a_1$ with conditional probability

$$\mathrm{P}(X > a_2 | X > a_1) = \frac{1 - F(a_2)}{1 - F(a_1)}. \tag{2.6}$$

Assume that decision instances are placed periodically after a constant time interval of length Δ, and that the components age during each interval. Consider a component whose age is a at a given decision instance. Its reliability, $R(\cdot)$, is the probability that it remains operational until the next instance, i.e.,

$$R(a) = \mathrm{P}(X > a + \Delta | X > a). \tag{2.7}$$

Another essential concept is the expected age of the component at the time of failure, conditioned on the event that the component stays operational until age a_1 but fails before age $a_2 > a_1$. In this case, the conditional CDF is

$$F_{X|a_1 \leq X \leq a_2}(x) = \frac{\mathrm{P}(X \leq x, a_1 \leq X \leq a_2)}{\mathrm{P}(a_1 \leq X \leq a_2)} = \begin{cases} 1, & \text{if } x > a_2 \\ \frac{F(x) - F(a_1)}{F(a_2) - F(a_1)}, & \text{if } a_1 \leq x \leq a_2 \\ 0, & \text{if } x < a_1. \end{cases} \tag{2.8}$$

Inserting this in (2.5) gives the conditional expected age at failure as

$$\mathbf{E}[X|a_1 \leq X \leq a_2] = a_1 + \int_{a_1}^{a_2} \left(1 - \frac{F(x) - F(a_1)}{F(a_2) - F(a_1)}\right) dx. \tag{2.9}$$

2.3 Maintenance modelling of multi-component systems

Multi-component systems (MCS) have $m > 1$ components which can be indexed with $i = 1, ..., m$. Typically, the maintenance costs of components are fixed. Component i can be maintained preventively before failure with cost c_i^p and correctively after failure with cost c_i^c. Commonly, it is assumed that preventive maintenance does not cost more than corrective maintenance: $c_i^p \leq c_i^c$ for all $i = 1, ..., m$ (e.g., Olde Keizer et al., 2018).

System failure can cause additional expenses. Recent discrete-time MDP models (e.g., Andersen et al., 2022) have accounted for system downtime with the concept of a *system failure cost*. This can be considered a conservative estimate of system failure, because the entire cost is incurred regardless of when, specifically, the system actually failed between the two maintenance instances (Zheng et al., 2023). A more accurate estimate of expected downtime costs is obtained with the *unit cost of downtime*. Still, using this approach requires that the expected time of system failure between maintenance instances is calculated (e.g., Xu et al., 2021).

Typically, components are assumed to have binary operational statuses, so that they can be modelled as either *operational* or *failed*. Even in this case, there are multiple ways to model component failures, depending on whether the operational time (often the age) or degradation level of a component is monitored. The dependence between operational time and failure likelihood can be modeled using the Weibull distribution (e.g., Vu et al., 2014). Its cumulative distribution function for $x \geq 0$ is

$$F(x) = 1 - e^{-(\frac{x}{\lambda})^k}, \tag{2.10}$$

where $\lambda > 0$ is the scale parameter and $k > 0$ the shape parameter. The Weibull distribution can be used to model various aging classes of life distributions, including increasing, decreasing, or constant failure rates with shape parameters $k > 1$, $k < 1$, and $k = 1$, respectively (Ahmad and Kamaruddin, 2012). The degradation of a component can be modelled, for example, with a gamma process (e.g., Andersen et al., 2022) or with a Poisson distribution (Olde Keizer et al., 2018). When the degradation level of a component is monitored, the component is considered to be failed when the degradation level exceeds a threshold. The continuous degradation of a component can be discretized so that a discrete-time MDP can be used for modelling (Andersen et al., 2022).

Positive economic dependency can be modeled with a *setup cost*, which is a fixed cost paid every time maintenance is performed, regardless of how many components are maintained simultaneously (Wildeman et al., 1997). Therefore, if t components are maintained simultaneously, the setup cost is saved $t-1$ times compared to the situation where such a setup cost would have to be paid for maintaining each component separately according to its optimal maintenance time.

Structural dependencies consist of *performance* and *technical* dependencies. Performance dependencies refer to how the system performs as a function of the operational statuses of its components. If the system has a reliability structure such that its components are in series, a single component failure causes the system to fail (e.g., Chalabi et al., 2016). If the components are in a parallel structure, the system is operational as long as at least one component is operational. With more complex systems, the structure can be any combination of series and parallel structures (e.g., Nguyen et al., 2015) or a k-out-of-N structure (e.g., Andersen et al., 2022).

In a series system in which components fail independently of each other, the reliability of the system between maintenance instances, $R_s(a_1, ..., a_m)$, given the ages of m components, is

$$R_s(a_1, ..., a_m) = \prod_{1 \leq i \leq m} R_i(a_i) \geq \rho, \tag{2.11}$$

where $R_i(a_i)$ comes from (2.7) and the reliability threshold ρ sets the minimum requirement for the reliability of the system.

2.3.1 Directed graphs

A directed graph representation (Diestel, 2010) can be used to model economic and technical dependencies between components. Formally, graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ is defined by

- the set of nodes $\mathcal{V}$ and
- the set of directed arcs $(x, y) \in \mathcal{A} \subset \mathcal{V} \times \mathcal{V}$ where each arc has a start node x and an end node y.

In this dissertation, the set of nodes is defined as $\mathcal{V} = \{0\} \cup \mathcal{D} \cup \mathcal{R}$, such that the node 0 is the *root node*. The sets $\mathcal{D}$ and $\mathcal{R}$ contain nodes that correspond to component disassembly and replacement actions, respectively. If the system has m components which are all replaceable, we have $|\mathcal{R}| = m$.

The root node is associated with the set-up cost. For each arc $(x, y) \in \mathcal{A}$, there is a cost weight c_{xy}, which represents the incremental cost of performing maintenance action y if the maintenance action x is also carried out during the same maintenance instance. The total cost of maintenance is determined with a tree called *a minimum-cost arborescence*, which is a connected subgraph of the graph and it (i) connects all action nodes corresponding to the maintenance actions scheduled to be performed simultaneously to the root node and (ii) minimizes the sum of the weights of the arcs included in the tree. This arborescence can be determined using Edmond's algorithm (Kleinberg and Tardos, 2006).

The graph allows the modelling of component-specific disassembly and replacement costs. Specifically, if a node $y \in \mathcal{D} \cup \mathcal{R}$ is connected directly to the root node, the corresponding maintenance action, i.e., either disassembly or replacement action, can be carried out without requiring other actions. Otherwise, the path from the root node to y passes through another node x, indicating a technical dependency. The action associated with node x must be executed before action y can be undertaken. Such dependencies arise from the fact that there can be components that have to be disassembled or replaced before other components can be maintained.

The graph structure allows for multiple paths from the root node to a single node, in which case the cost of maintaining a component may depend on which other components are maintained simultaneously. The directed graph can also be modelled as a disassembly matrix (e.g., Dinh et al., 2024). Technical dependencies can also be examined from the perspective of minimizing the costs of the disassembly sequence (e.g., Zhou et al., 2015).

An illustrative example of this graph-based structure is shown in Figure 2.1, which represents a ground transportation equipment with four components: engine 1 (E1), engine 2 (E2), chassis (C), and wheels (W). The disassembly decision nodes are $\mathcal{D} = \{\mathrm{DE1}, \mathrm{DE2}, \mathrm{DE12}, \mathrm{DC}, \mathrm{DW}\}$ and replacement decision nodes $\mathcal{R} = \{\mathrm{E1}, \mathrm{E2}, \mathrm{C}, \mathrm{W}\}$. Every component needs to be disassembled before it can be replaced. Additionally, both engines have to be disassembled (node DE12) before the chassis can be disassembled. Before the wheels can be disassembled, the other three components must be disassembled.

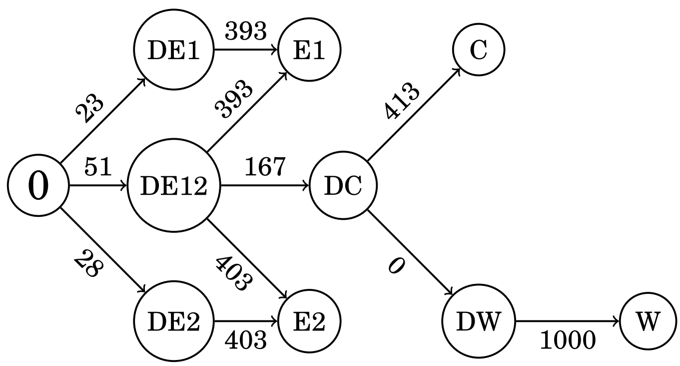

Figure 2.1. Graph of technical dependencies of a four-component system. Reproduced from Paper I (Leppinen et al., 2025).

2.3.2 Binary decision diagrams

Binary decision diagrams can be used to model a broad range of reliability structures, based on the logical dependencies in a coherent system whose binary operational status can be expressed as a function of the binary statuses of its constituent components. For illustration, consider a four-component system with the following reliability structure: engines one (E1) and two (E2) are in parallel, and both the chassis (C) and wheels (W) are in series with the engines. The reliability structure and a corresponding binary decision diagram are in Figure 2.2. The system fails when both engines, the chassis, or the wheels fail.

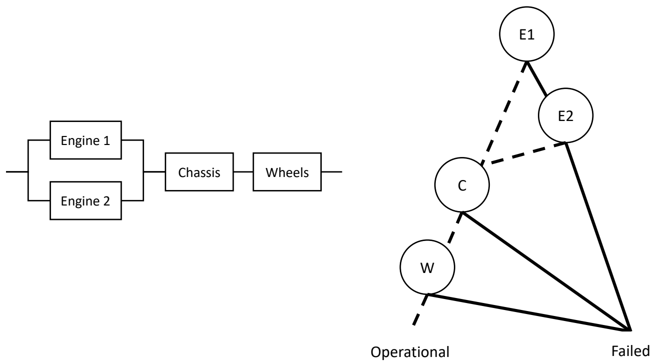

Figure 2.2. Reliability structure of the system (left panel) and the corresponding binary decision diagram (right panel)

For a given system, there can be alternative yet logically equivalent binary decision diagram representations. In the binary decision diagram, the left (dashed) branch after the component represents the operational status of a component, and the right (solid) branch represents the failed status. The paths between components end in either an operational or failed status of the system at the bottom of the diagram. From node E1, there are two paths including dashes to an operational system, as shown in Figure 2.2. The reliability of a component is the probability that the component remains operational, i.e., the probability of the left branch. Correspondingly, a component fails, i.e., branches to the right, with probability $1 - R(a)$. If the components fail independently, the probability of a path is the product of the probabilities of the branches that form the path. Since paths are mutually exclusive and collectively exhaustive, the reliability of the system is the sum of the probabilities of those paths that correspond to a system that operates. For example, the reliability of the system in Figure 2.2 is

$$\begin{aligned} R_s(a_{\mathrm{E1}}, a_{\mathrm{E2}}, a_{\mathrm{C}}, a_{\mathrm{W}}) &= R_{\mathrm{E1}}(a_{\mathrm{E1}})R_{\mathrm{C}}(a_{\mathrm{C}})R_{\mathrm{W}}(a_{\mathrm{W}}) \\ &\quad + (1 - R_{\mathrm{E1}}(a_{\mathrm{E1}}))R_{\mathrm{E2}}(a_{\mathrm{E2}})R_{\mathrm{C}}(a_{\mathrm{C}})R_{\mathrm{W}}(a_{\mathrm{W}}). \end{aligned} \tag{2.12}$$

2.4 Stage-gate process

The stage-gate process is a framework for new product development, spanning from the initial idea to market launch (Cooper, 1990). The development work is conducted in stages, which are separated by gates at which decisions about the continuation of the development work are made. For each gate, a set of deliverables is specified. Deliverables are produced for each product idea to help assess the quality criteria that the idea must meet before its development can proceed to the next stage. Typically, in the early stages, quality criteria are largely qualitative and address "must meet" and "should meet" issues. In the later stages, when more details are available, quantitative criteria, such as marketing budgets, can be considered. Decisions at the gates are typically made by senior managers who possess an adequate professional background and the authority to approve the necessary resources for development. This promotes managerial involvement and commitment to support the development process.

In the initial screening stages, there are usually multiple new product ideas. When these ideas are assessed against the evaluation criteria, multi-criteria decision-making methods can be helpful at the gates to determine whether to continue or terminate the development of each idea. There is an extensive range of relevant methods, such as the Analytical Hierarchy Process (Saaty, 1994) and Multi-Attribute Value Theory (Keeney and Raiffa, 1993). However, if there are uncertainties about how the items under development, such as ideas or projects, perform with regard to the criteria or how to prefer one criterion over another, methods developed for preference programming can be used. For example, Robust Portfolio Modeling (Liesiö et al., 2007) is one such method. These methods help to determine robust decision recommendations, which are based on the explicit recognition of uncertain information of the projects.

3. Contributions

This dissertation develops methods for improving the maintenance management of technical systems. Papers I-II contribute to the modelling of structural dependencies of multi-component systems (MCS), the formulation of the maintenance scheduling problem using Markov decision processes (MDP), and the derivation of optimal maintenance scheduling policies. Paper III contributes to the literature of decision processes by proposing a structured approach for guiding the development of an artificial intelligence (AI)-based solution for the prognostic and health management (PHM) task. Table 3.1 summarizes the specific contributions of the three papers.

3.1 Paper I

Maintenance scheduling for MCSs with dependencies has received plenty of attention in the literature (Olde Keizer et al., 2017; de Jonge and Scarf, 2020). However, some aspects of the problem, such as technical dependencies combined with a system reliability requirement, have received little attention. The first of the three main contributions of Paper I is that the proposed approach helps determine optimal maintenance policies that satisfy reliability requirements in the presence of constraints arising from technical dependencies. Towards this end, it introduces directed graphs as a powerful and intuitively understandable tool to represent the economic and technical dependencies of the MCS. Second, the paper formulates the problem as an MDP model. Third, it solves the model using the modified policy-iteration algorithm, which is accelerated using Anderson acceleration, thereby expanding the set of value vector update schemes of the earlier numerical studies (Andersen et al., 2022).

Paper	Objectives	Methodologies	Observations on results
I: An optimization model for determining cost-efficient maintenance policies for multi-component systems with economic and structural dependencies	Model optimal maintenance scheduling for MCS under periodic maintenance, reliability constraints, and both economic and structural dependencies.	Directed graphs, MDP, modified policy iteration	The optimal policy outperforms simple heuristics. MDPs with large state spaces can be solved reasonably fast.
II: Balancing downtime and maintenance costs for multi-component systems with economic and structural dependencies	Model periodic maintenance scheduling for an MCS to minimize expected maintenance and downtime costs under economic and structural dependencies.	Directed graphs, binary decision diagrams, MDP, modified policy iteration	The unit cost of downtime impacts the structure of the optimal policy. The model helps compare system designs in terms of expected downtime or maintenance costs.
III: A stage-gate decision process for guiding the development of AI solutions for preventive maintenance	Elaborate a structured decision process for guiding the development of an AI-based solution for the PHM task.	Stage-gate process, Robust Portfolio Modeling	The process provides a consistent evaluation framework for development and helps mitigate the risk of missing development objectives due to premature fixation on a single candidate solution.

Table 3.1. Summary of publications

3.2 Paper II

The failure of an MCS causes downtime if immediate corrective maintenance is not possible. To evaluate the costs caused by such downtime, the expected downtime needs to be estimated in a periodic maintenance setting. Xu et al. (2021) and Zheng et al. (2023) compute this for MCSs where k-out-of-N components needs to operate for the system to operate.

Paper II generalizes earlier results by determining expected downtime for any binary-valued reliability structure. Furthermore, the computational effort required to solve the optimal policy is reduced by identifying state transitions that have the same expected downtime. Thus, the expected downtime for those state transitions can be obtained with a single calculation. Finally, the optimal policy's performance, measured by expected costs and downtime, is assessed for different system structures and their associated structural dependencies. This contrasts with earlier papers in which optimal policies are typically solved for a fixed set of structural dependencies, such as k-out-of-N performance dependencies (Zheng et al., 2023).

3.3 Paper III

Many AI-based data-driven algorithms can be relied on for PHM (Atamuradov et al., 2017). However, the process of building on AI-based algorithms to develop solutions to solve a given PHM task with available data resources has not received attention in the literature on decision processes. In response to this recognition, Paper III develops a stage-gate process to guide the development of an AI-based solution that supports PHM. For each decision gate, the requested deliverables of solution candidates are formed and evaluated based on a set of criteria. In total, 17 criteria are proposed, each of which is linked to one of the six overarching development objectives. These criteria encompass both qualitative and quantitative aspects. Likert scales (Likert, 1932) are used to evaluate qualitative criteria. At decision gates, the decision to continue or terminate the development of candidates is supported by Robust Portfolio Modeling (Liesiö et al., 2007, 2008), which admits incomplete evaluation scores and preferences information. Finally, the evaluation of candidates is based on hierarchical weighting, allowing the decision-maker to state their preferences separately between criteria linked to a single development objective and between objectives.

4. Discussion

This dissertation extends the range of analytical approaches that can be employed to improve the maintenance management of technical systems. Papers I and II develop novel Markov decision process (MDP) formulations for solving a periodic maintenance schedule for multi-component systems (MCSs) under economic, technical, and performance dependencies and a reliability requirement for system operation. Paper III develops a stage-gate model to guide the development of artificial intelligence (AI)-based methods for supporting prognostics and health management (PHM).

Maintenance management includes decision-making at operational, tactical, and strategic levels (Ben-Daya et al., 2016). The model of Paper I can support decision-making at all these levels. At the operational level, the recommended policy identifies optimal replacement decisions, regardless of the state of the system, as long as the reliability threshold is satisfied. The simulation of the policy supports tactical planning, including the computation of the required number of spare components and the estimation of how often available maintenance opportunities are actually used to perform maintenance actions. The reliability threshold and the distance between periodic maintenance instances can be adjusted according to strategic goals set by the maintenance manager.

Models for optimal maintenance scheduling of MCSs typically aim to minimize maintenance expenses, ensure sufficient system reliability, and minimize system downtime. In the literature, only a few studies have considered these objectives simultaneously, typically by treating some of them as objectives and others as constraints (see, e.g., Dao and Zuo, 2017; Dhanisetty et al., 2018; Wang et al., 2022). Paper II addresses all three objectives in periodic maintenance, while allowing for the modeling of positive economic dependence and various technical and performance dependencies.

When developing new solutions to problems from initial ideas, it is often uncertain which one will yield the most desirable outcome (Vilkkumaa et al., 2015). Uncertainty can be reduced by gathering evidence using available resources. Paper III introduces a process to help address the uncertainties associated with developing AI solutions for PHM tasks and to distribute resources effectively. The process has similarities with the new product development process (Cooper, 1990). Still, one of the main differences is that developed AI-based solutions are case-specific to a single PHM task and for in-house use only.

This dissertation opens up several avenues for research. Preventive maintenance involves uncertainties, including various causes of failure, the competence of the maintenance worker, variations in maintenance tools and equipment, and complex operational conditions. Some of these factors remain relevant when components are replaced with new ones (Xu et al., 2021). The MDP models of Papers I and II could be expanded to model these phenomena, for example, by assigning a random age other than zero to the replaced components.

Additionally, the performance of the Anderson accelerated Gauss-Seidel modified policy iteration algorithm could be studied further. Its convergence depends on the order in which the components of the value vector are updated, the initial guess of the value vector, the number of value computations, and the size of the memory used during Anderson acceleration (Geist and Scherrer, 2018). Defining limits in terms of state-space size or number of system components would determine how large problems can be solved with the algorithm.

Although models in Papers I and II are built using time-based maintenance depending on component ages, they can be applied to condition-based maintenance with minor modifications. For example, failure probabilities during state transitions for some components can be determined from the deterioration that is governed by a stochastic process (Do and Bérenguer, 2020). Then, the maintenance of the system can be approached as a combination of time-based and condition-based maintenance. Some components are maintained based on age, while others are maintained based on their condition. Using such an approach can identify situations where monitoring the condition of a specific component is unnecessary, as it may be more effective to combine its maintenance with either age-based or condition-based maintenance of another component, especially if there are dependencies between components that call for simultaneous maintenance. Thus, the models in Papers I and II help analyze the importance of monitoring the condition of different system components, which supports PHM implementation (Atamuradov et al., 2017).

The stage-gate model of Paper III guides the development of an AI-based solution for a given decision context characterised by the availability of data, performance requirements, and the PHM task. Future research could investigate whether the properties of the PHM task can be harnessed to generate recommendations about which criteria and objectives, in particular, should be used and how to elicit preferences concerning these criteria and objectives. One could also investigate how the development process could help determine the data qualities that best support the completion of the PHM task. Identifying possible requirements for data quality can help in selecting sensors for condition monitoring, and thus support PHM implementation (Atamuradov et al., 2017).

References

Hasnida Ab-Samat and Shahrul Kamaruddin. Opportunistic maintenance (OM) as a new advancement in maintenance approaches: a review. *Journal of Quality in Maintenance Engineering*, 20(2):98–121, 2014.

Rosmaini Ahmad and Shahrul Kamaruddin. An overview of time-based and condition-based maintenance in industrial application. *Computers & Industrial Engineering*, 63(1):135–149, 2012.

Dawn An, Nam H Kim, and Joo-Ho Choi. Practical options for selecting data-driven or physics-based prognostics algorithms with reviews. *Reliability Engineering & System Safety*, 133:223–236, 2015.

Jesper Fink Andersen, Anders Reenberg Andersen, Murat Kulahci, and Bo Friis Nielsen. A numerical study of Markov decision process algorithms for multi-component replacement problems. *European Journal of Operational Research*, 299(3):898–909, 2022.

Vepa Atamuradov, Kamal Medjaher, Pierre Dersin, Benjamin Lamoureux, and Noureddine Zerhouni. Prognostics and health management for maintenance practitioners-review, implementation and tools evaluation. *International Journal of Prognostics and Health Management*, 8(3):1–31, 2017.

Richard E Barlow and Frank Proschan. *Mathematical theory of reliability*. Society for Industrial and Applied Mathematics, 1996.

Mohamed Ben-Daya, Uday Kumar, and DN Prabhakar Murthy. *Introduction to maintenance engineering: modelling, optimization and management*. John Wiley & Sons, 2016.

Maurizio Bevilacqua and Marcello Braglia. The analytic hierarchy process applied to maintenance strategy selection. *Reliability Engineering & System Safety*, 70 (1):71–83, 2000.

Keomany Bouvard, Samuel Artus, Christophe Bérenguer, and Vincent Cocquempot. Condition-based dynamic maintenance operations planning & grouping. application to commercial heavy vehicles. *Reliability Engineering & System Safety*, 96(6):601–610, 2011.

Nadia Chalabi, Mohammed Dahane, Bouziane Beldjilali, and Abdelkader Neki. Optimisation of preventive maintenance grouping strategy for multi-component series systems: particle swarm based approach. *Computers & Industrial Engineering*, 102:440–451, 2016.

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

Robert G Cooper. Stage-gate systems: a new tool for managing new products. *Business Horizons*, 33(3):44–54, 1990.

Ross B Corotis, J Hugh Ellis, and Mingxiang Jiang. Modeling of risk-based inspection, maintenance and life-cycle cost with partially observable Markov decision processes. *Structure and Infrastructure Engineering*, 1(1):75–84, 2005.

Cuong D Dao and Ming J Zuo. Selective maintenance of multi-state systems with structural dependence. *Reliability Engineering & System Safety*, 159:184–195, 2017.

Bram de Jonge and Philip A Scarf. A review on maintenance optimization. *European Journal of Operational Research*, 285(3):805–824, 2020.

Rommert Dekker. Applications of maintenance optimization models: a review and analysis. *Reliability Engineering & System Safety*, 51(3):229–240, 1996.

VS Viswanath Dhanisetty, WJC Verhagen, and Richard Curran. Multi-criteria weighted decision making for operational maintenance processes. *Journal of Air Transport Management*, 68:152–164, 2018.

Reinhard. Diestel. *Graph theory*. Graduate texts in mathematics, 173. Springer, Heidelberg, 4th ed. edition, 2010. ISBN 3-642-14278-8.

Duc-Hanh Dinh, Phuc Do, and Benoit Iung. Degradation modeling and reliability assessment for a multi-component system with structural dependence. *Computers & Industrial Engineering*, page 106443, 2020.

Duc-Hanh Dinh, Phuc Do, Benoit Iung, and Pham-The-Nhan Nguyen. Reliability modeling and opportunistic maintenance optimization for a multicomponent system with structural dependence. *Reliability Engineering & System Safety*, 241:109708, 2024.

Phuc Do and Christophe Bérenguer. Conditional reliability-based importance measures. *Reliability Engineering & System Safety*, 193:106633, 2020.

Amik Garg and SG Deshmukh. Maintenance management: literature review and directions. *Journal of Quality in Maintenance Engineering*, 12(3):205–238, 2006.

Matthieu Geist and Bruno Scherrer. Anderson acceleration for reinforcement learning. *arXiv preprint arXiv:1809.09501*, 2018.

Ronald A Howard. *Dynamic programming and Markov processes.* John Wiley, 1960.

Andrew KS Jardine, Daming Lin, and Dragan Banjevic. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20(7):1483–1510, 2006.

Ralph L Keeney and Howard Raiffa. *Decisions with multiple objectives: preferences and value trade-offs*. Cambridge University Press, 1993.

Samir Khan, Seiji Tsutsumi, Takehisa Yairi, and Shinichi Nakasuka. Robustness of AI-based prognostic and systems health management. *Annual Reviews in Control*, 51:130–152, 2021.

Jon Kleinberg and Eva Tardos. *Algorithm design*. Pearson Education India, 2006.

Yaguo Lei, Naipeng Li, Liang Guo, Ningbo Li, Tao Yan, and Jing Lin. Machinery health prognostics: a systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*, 104:799–834, 2018.

Jussi Leppinen, Antti Punkka, Tommi Ekholm, and Ahti Salo. An optimization model for determining cost-efficient maintenance policies for multi-component systems with economic and structural dependencies. *Omega*, 130:103162, 2025.

Gregory Levitin, Liudong Xing, and Yuanshun Dai. Minimum downtime operation and maintenance scheduling for resource-constrained system. *Reliability Engineering & System Safety*, 238:109465, 2023.

Juuso Liesiö, Pekka Mild, and Ahti Salo. Preference programming for robust portfolio modeling and project selection. *European Journal of Operational Research*, 181(3):1488–1505, 2007.

Juuso Liesiö, Pekka Mild, and Ahti Salo. Robust portfolio modeling with incomplete cost information and project interdependencies. *European Journal of Operational Research*, 190(3):679–695, 2008.

Rensis Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 1932.

Kim-Anh Nguyen, Phuc Do, and Antoine Grall. Multi-level predictive maintenance for multi-component systems. *Reliability Engineering & System Safety*, 144:83–94, 2015.

Robin P Nicolai and Rommert Dekker. Optimal maintenance of multi-component systems: a review. In *Complex system maintenance handbook*, pages 263–286. Springer, 2008.

Minou CA Olde Keizer, Simme Douwe P Flapper, and Ruud H Teunter. Condition-based maintenance policies for systems with multiple dependent components: a review. *European Journal of Operational Research*, 261(2):405–420, 2017.

Minou CA Olde Keizer, Ruud H Teunter, Jasper Veldman, and M Zied Babai. Condition-based maintenance for systems with economic dependence and load sharing. *International Journal of Production Economics*, 195:319–327, 2018.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Thomas L Saaty. *Fundamentals of decision making and priority theory with the analytic hierarchy process*. RWS publications, 1994.

David Sherwin. A review of overall models for maintenance management. *Journal of Quality in Maintenance Engineering*, 6(3):138–164, 2000.

Eeva Vilkkumaa, Ahti Salo, Juuso Liesiö, and Afzal Siddiqui. Fostering breakthrough technologies — how do optimal funding decisions depend on evaluation accuracy? *Technological Forecasting and Social Change*, 96:173–190, 2015.

Hai Canh Vu, Phuc Do, Anne Barros, and Christophe Bérenguer. Maintenance grouping strategy for multi-component systems with dynamic contexts. *Reliability Engineering & System Safety*, 132:233–249, 2014.

Yukun Wang, Xiaopeng Li, Junyan Chen, and Yiliu Liu. A condition-based maintenance policy for multi-component systems subject to stochastic and economic dependencies. *Reliability Engineering & System Safety*, 219:108174, 2022.

Ralph Edwin Wildeman, Rommert Dekker, and ACJM Smit. A dynamic policy for grouping maintenance activities. *European Journal of Operational Research*, 99(3):530–551, 1997.

Jun Xu, Zhenglin Liang, Yan-Fu Li, and Kaibo Wang. Generalized condition-based maintenance optimization for multi-component systems considering stochastic dependency and imperfect maintenance. *Reliability Engineering & System Safety*, 211:107592, 2021.

Xiujie Zhao, Piao Chen, and Loon Ching Tang. Condition-based maintenance via Markov decision processes: a review. *Frontiers of Engineering Management*, pages 1–13, 2025.

Meimei Zheng, Jie Lin, Tangbin Xia, Yu Liu, and Ershun Pan. Joint condition-based maintenance and spare provisioning policy for a K-out-of-N system with failures during inspection intervals. *European Journal of Operational Research*, 308(3):1220–1232, 2023.

Xiaojun Zhou, Kaimin Huang, Lifeng Xi, and Jay Lee. Preventive maintenance modeling for multi-component systems with considering stochastic failures and disassembly sequence. *Reliability Engineering & System Safety*, 142:231–237, 2015.

Business, Economy

Art, Design, Architecture

Science, Technology

Crossover

Doctoral Theses

Aalto DT 50/2026

ISBN 978-952-64-3008-9
ISBN 978-952-64-3007-2 (pdf)

Aalto University
School of Science
Department of Mathematics and
System Analysis
aalto.fi