Maintenance optimization for multi-component systems using Markov decision processes

Andersen, Jesper Fink

Publication date:
2022

Document Version
Publisher's PDF, also known as Version of record

Ph.D. Thesis
Doctor of Philosophy

DTU Compute
Department of Applied Mathematics and Computer Science

Maintenance optimization for multi-component systems using Markov decision processes

Jesper Fink Andersen

Kongens Lyngby 2021

DTU Compute
Department of Applied Mathematics and Computer Science
Technical University of Denmark

Richard Petersens Plads
Building 324
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

Summary

Modern engineering systems are often comprised of multiple components that deteriorate with use. In a civil structure, this deterioration will, over time, lead to an unacceptable risk of failure. In manufacturing and production systems, component failures caused by deterioration lead to unforeseen downtime. Since more manufacturing processes are being automated and the requirements for safety become stricter, the maintenance function accounts for an increasingly larger fraction of the total operational costs for such systems. An effective maintenance policy is, therefore, necessary to compensate for this trend.

Both industry and academia are now mostly focused on a maintenance paradigm called Condition-Based Maintenance (CBM), where maintenance activities are carried out based on the monitored deterioration state of the system. In a more traditional approach, Time-Based Maintenance (TBM), the next maintenance activity is scheduled based on the elapsed time since the previous maintenance activity without consideration to the condition of the system at the scheduled time. By monitoring the condition of the system, the CBM approach can potentially reduce the number of redundant maintenance activities and unforeseen failures.

In this dissertation, we consider the problem of maintenance optimization in multi-component systems, both for the TBM and the CBM approach. The dissertation is divided into six chapters, three of which are academic paper manuscripts.

Many heuristic maintenance policies have been developed for multi-component systems. We focus on optimal maintenance policies identified using Markov Decision Process (MDP) models and dynamic programming optimization algorithms. Identifying the optimal policy in a general MDP model is computationally demanding if the state space has multiple dimensions. In paper A, we use one MDP state dimension for each component in a unifying model framework for the TBM and CBM approach in a multi-component system. We then perform numerical experiments to determine the practical computational size limit of the MDP, that is, the largest number of components in the system, for which we can obtain an optimal maintenance policy.

In Paper B, we consider a CBM system with continuously deteriorating components and investigate the effects of discretization, which is a necessary step for dynamic programming optimization. We compare different methods for discretization and demonstrate that a relatively coarse discretization still results in a near-optimal maintenance policy. Even though the discretization is primarily a technical matter pertaining to the optimization procedure, we can also draw a connection between

the results from one of the tested discretization methods and a common practice of classifying the system condition on a discrete scale.

Many companies are currently seeking to improve their maintenance practices by implementing CBM. The rationale is that CBM is more cost-efficient than TBM, because maintenance can be performed just in time before a component fails in CBM. In Paper C, we quantify this benefit by comparing the performance of optimal TBM and CBM policies for a multi-component system. Specifically, we show how changing the number of components and the degree of stochastic and economic dependence between components affect the difference between the performance of the TBM and CBM policies.

Summary (Danish)

Moderne tekniske systemer består ofte af flere komponenter, som slides ved brug. I en konstruktion vil sådan slitage over tid fører til en uacceptabel risiko for en defekt. I fremstillings- og produktionssystemer fører komponentfejl forårsaget af slitage til uforudset nedetid. Da flere og flere fremstillingsprocesser automatiseres, og kravene til sikkerhed bliver strengere, udgør vedligehold en stadig større andel af de samlede driftsomkostninger for sådanne systemer. For at kompensere for denne tendens er et effektivt vedligeholdelsesprogram nødvendigt.

Både industrien og den akademiske verden er i denne tid mest fokuseret på et vedligeholdelsesparadigme kaldet tilstandsbaseret vedligehold (CBM), hvor vedligehold udføres baseret på systemets overvågede fysiske tilstand. I en mere traditionel tilgang, tidsbaseret vedligehold (TBM), planlægges den næste vedligeholdelsesopgave baseret på den forløbne tid siden sidste vedligeholdelsesopgave uden at tage hensyn til systemets tilstand på det planlagte tidspunkt. Ved at overvåge systemets tilstand løbende kan CBM reducere antallet af overflødige vedligeholdelsesopgaver og spontane fejl.

Denne afhandling omhandler vedligeholdelsesoptimering i flerkomponentsystemer, både for TBM- og CBM-tilgangen. Afhandlingen er opdelt i seks kapitler, hvoraf tre er manuskripter til videnskabelige artikler.

Der er blevet udviklet mange heuristiske vedligeholdelsesprogrammer til flerkomponentsystemer, men vi fokuserer på optimale vedligeholdelsesprogrammer, som udregnes ved brug af Markov beslutningsprocesser (MDP) og dynamisk programmering. At finde den optimale løsning i en generel MDP er beregningsmæssigt krævende, hvis tilstandsrummet har mange dimensioner. I artikel A bruger vi én dimension i MDP tilstandsrummet for hver komponent i en samlet model for TBM- og CBM-tilgangen af et flerkomponentsystem. Herefter fremlægger vi numeriske eksperimenter, der bestemmer den praktiske beregningsmæssige øvre grænse for MDP'en, det vil sige antallet af komponenter i systemet, hvor vi stadig kan beregne et optimal vedligeholdelsesprogram.

I artikel B betragter vi et CBM flerkomponentsystem, hvor komponenterne følger en kontinuert slitageproces, og vi undersøger effekterne af diskretisering, som er nødvendigt for at kunne optimere med dynamisk programmering. Vi sammenligner forskellige metoder til diskretisering og viser, at en relativt grov diskretisering stadig resulterer i et næsten optimalt vedligeholdelsesprogram. Diskretiseringen er umiddelbart kun en beregningsmæssig nødvendighed for optimeringsalgoritmen, men vi

relaterer også resultaterne fra en af de afprøvede diskretiseringsmetoder til en almindelig praksis, hvor systemtilstanden klassificeres på en diskret skala.

Mange virksomheder forsøger i øjeblikket at forbedre deres vedligeholdelsesprocesser ved at implementere CBM. Rationalet er, at CBM er mere omkostningseffektivt end TBM, fordi vedligeholdelse kan udføres umiddelbart før en komponent fejler i CBM. I artikel C kvantificerer vi denne forskel ved at sammenligne omkostningen ved at bruge optimale TBM- og CBM-programmer for et flerkomponentsystem. Mere specifikt viser vi, hvordan ændringer i antallet af komponenter og graden af stokastisk og økonomisk afhængighed mellem komponenterne påvirker forskellen mellem omkostningen af TBM of CBM vedligeholdelsesprogrammerne.

Preface

This thesis has been prepared at the department of Applied Mathematics and Computer Science, Section for Statistics and Data Analysis at the Technical University of Denmark in fulfillment of the requirements for acquiring a Ph.D. degree in applied mathematics and computer science.

The Ph.D. project was funded by the Technical University of Denmark and has been completed under the guidance of Professor Bo Friis Nielsen (main supervisor) and Associate professor Murat Kulahci (co-supervisor).

Kongens Lyngby, October 1, 2021

Jesper Fink Andersen

Acknowledgements

To be honest, I am writing this at 11:30 pm; 30 minutes before I hand in the thesis. I apologize to all the people I mention below, if my gratitude towards you does not come across very eloquently, but I am unfortunately a little tired right now.

First of all, I would like to thank my supervisors, Bo Friis Nielsen and Murat Kulahci. Besides always giving me thorough and honest feedback on my work, they are excellent at giving pep-talks, which were very much needed during the last six months of the Ph.D. project. I have learned a lot from both of you, and I appreciate how much we have laughed during our weekly supervisor meetings, from which I have always left in good spirits. I would also like to thank Anders Reenberg Andersen, for sharing my enthusiasm for MDPs. It was very nice to have someone to discuss the nerdy technical details with.

Writing a PhD thesis is a solitary journey, and a lockdown of society due to a global pandemic does not make the journey any less lonely. Therefore, I am lucky to have had so much support from my soon-to-be wife, Dina, especially during the times when feelings of meaninglessness and self-doubt were strong. My soon-to-be mother-in-law also deserves a special thanks for cheering me on from the sideline, and also for walking our dog every day those couple of weeks when Dina had broken her foot, so I could concentrate fully on revising a paper.

Acronyms

ADP Approximate Dynamic Programming. 21

AR Age-Replacement. 36

BR Block-Replacement. 25, 41

CBM Condition-Based Maintenance. 8

CM Corrective Maintenance. 124

CMDP Constrained Markov Decision Process. 20

CTMC Continuous-Time Markov Chain. 20

DP Dynamic Programming. 5

DTMC Discrete-Time Markov Chain. 14

MDP Markov Decision Process. 5

MILP Mixed Integer Linear Programming. 35

PI Policy Itereation. 16

PIR Periodic Inspection and Replacement. 36

PM Preventive Maintenance. 124

POMDP Partially Observable Markov Decision Process. 7

RL Reinforcement Learning. 21

SMDP Semi-Markov Decision Process. 20

TBM Time-Based Maintenance. 8

VI Value Itereation. 16

Contents

CHAPTER 1

Introduction

In manufacturing, many process improvement efforts have aimed at increasing quality and reducing waste. In a Six Sigma process for example the requirement is to observe no more than 3.4 defective parts per million parts produced (Montgomery and Woodall 2008). When such a level of precision and high quality is achieved, increasing the production rate is the primary way to ensure competitiveness. One of the roadblocks in this pursuit is the excessive system downtime. In a survey of Swedish manufacturing companies the estimated downtime costs amount to 23.9% of the total cost of manufacturing on average (Tabikh 2014). An effective strategy for the maintenance of equipment is critical for all production companies that aim to stay competitive. In 2016, U.S. census data estimated that companies spent $50 billion dollars on outsourced maintenance and repair work, excluding the internal expenditures on labor and materials (Thomas 2018). In Mobley (2002) it is stated that the U.S. industry spends more than $200 billion on maintenance of production plant facilities each year, while one third of this is wasted on unnecessary or improperly carried out activities. In the chemical industry, the amount of maintenance personnel can be as high as 30% of the total work force (Waeyenbergh and Pintelon 2002). In a case study of an Italian oil refinery, Bevilacqua and Braglia (2000), it is estimated that maintenance department costs account for between 15% and 70% of the total production cost. The operation and maintenance costs of a wind turbine can account for 75%-90% of the investment cost according to Vachon (2002). In the maritime sector, maintenance and repair of bulk carriers account for 40% of the operating costs (Eruguz et al. 2017). The maintenance expenditures for engineering structures such as roads, bridges and railways were predicted by Dekker and Scarf (1998) to be continuously increasing, due to higher performance requirements and outsourcing of maintenance. Figure 1.1 shows the government financed maintenance expenditure on road infrastructure for various countries relative to the year 1998. Indeed, these numbers indicate the prediction of Dekker and Scarf (1998) was generally accurate. Regarding outsourcing, Bowman and Schmee (2001) consider a maintenance service provider for aircraft engines, where a typical maintenance contract for a fleet of aircrafts span many years and the total costs for the service provider exceed a billion dollars. The magnitude and uncertainty of the costs could be a significant financial risk for the service provider, and in Bowman and Schmee (2001) a mathematical model is developed for pricing the contracts in order to mitigate this risk.

Besides the references already mentioned, we were not able to find any concrete

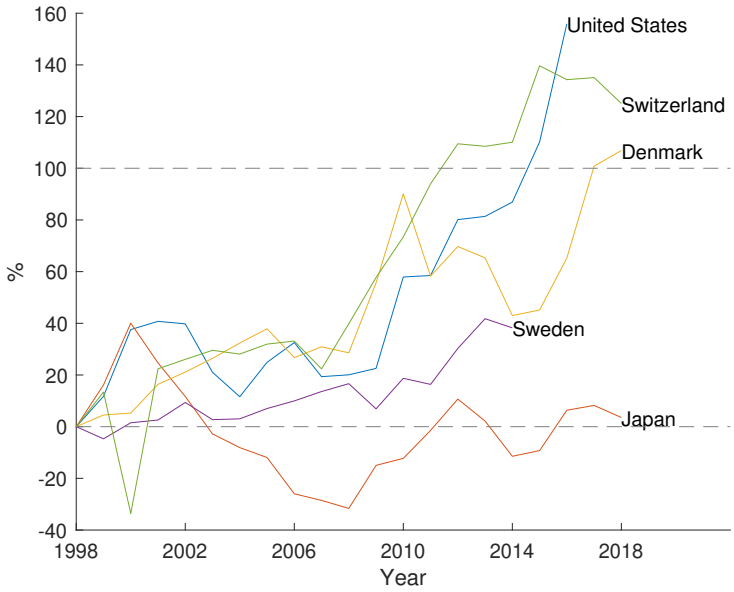


Figure 1.1: Increase in public expenditure on road infrastructure maintenance relative to the year 1998. Source: ITF (2021).

numbers on costs of maintenance in industry. Even though, some of the references are old, they still illustrate the vast economic scales. During this PhD project, we have also been in contact with several large companies, all of which are currently invested in improving their maintenance processes. Altogether, this underlines the importance of mathematical models for maintenance optimization, which is the topic of this thesis.

The field of maintenance optimization is very diverse and relates to many academic fields ranging from mathematical disciplines such as statistics, econometrics, stochastic modeling, operations research, and machine learning to domain specific field such as electrical, mechanical, and civil engineering. In this thesis we mainly consider the field of operations research, and more specifically, sequential decision-making under uncertainty. It is easy to see why the latter subject plays a particularly large role within maintenance literature. We perform maintenance on equipment in order to extend its life and keep it from failing. The uncertainty comes into play, because we never know with certainty how much we extend the life or when in the future the equipment will fail. The sequential decision-making part is because we typically perform multiple maintenance activities throughout the period, in which we need the equipment, and the decisions then concern the "when" and "how" we perform maintenance.

1.1 A generic example

There are many different approaches to maintenance optimization, but they all share some of the same characteristics. Let us first consider a minimal example that allows us to identify six key aspects that are common for the majority of maintenance optimization problems.

We consider a physical **system**, which is continuously operating, and we wish to keep it in an operating state. Very often the system is a mechanical machine, but it may also be a civil structure or an electronic device. In the context of a production assembly line, the system could be any part of the line that would in some way inhibit the production if it fails. It is common to only consider part of a system, for instance a single critical component. The manner and frequency, by which the system is observed, are also important aspects that influence how the system should be modeled. We might have imperfect or no incoming information about the state of the system. The system state can also have multiple dimensions, for instance if the system consists of multiple components.

The system is subject to **deterioration** and if it is left to itself, it will ultimately fail and stop operating. There are many different ways of modeling deterioration depending on the type of system and the nature of the failure. If the deterioration is gradual, then a stochastic process with a continuous state space, like the one illustrated in Figure 1.2, is usually appropriate.

The deterioration state of the system is usually not known by the decision-maker at all times. It might be that the state can only be observed by stopping the equipment and inspecting its components. Other times, it may not even be possible to observe the deterioration state, but only whether or not the system is still functioning. The **information** about the system state available to the decision-maker is therefore also important to consider in a maintenance model. For the current example, we assume the system is inspected periodically with intervals of unit length, and that inspections reveal the condition perfectly, i.e., the level of the stochastic process.

To keep the system from failing we must perform **maintenance** on the system, in order to improve its condition. In the current example, let us assume that we only use the simplest and most extreme form of maintenance, namely replacement of the entire system with a new and identical one. At each inspection, we then decide if a replacement is needed or not.

The underlying reason we want to keep the system operating is because we want to maximize a certain **utility**. Usually this is formulated in terms of minimizing maintenance costs. Let us assume that it is cheaper to replace the system before it fails, than after it fails. There are many situations where this could be the case. For instance, if the system is connected to other systems that are also damaged by the failure, these could need maintenance as well. Very often the corrective cost also reflects a system downtime cost because of loss of production or availability. For safety critical systems, a minimum required reliability may also be incorporated in the utility as a constraint that conflicts with maintenance cost minimization.

This brings to the final aspect, which is the **optimization** part. Every main-

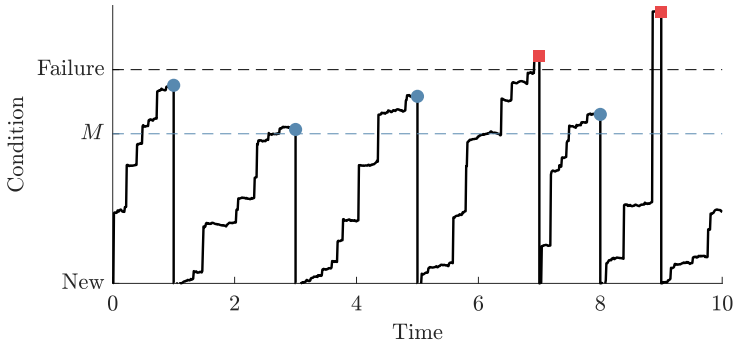


Figure 1.2: Graphical depiction of a generic maintenance optimization problem. The gradually worsening condition of the system is represented by the solid line. The system is inspected at times $1, 2, 3, \ldots$ and preventive replacements (blue circles) are performed when the system is found above the threshold, M. The system fails when the condition exceeds the failure threshold and is then correctively replaced (red squares).

tenance policy has an associated utility, and we want to identify the policy that maximizes this utility. For the current example, a good maintenance policy is one that balances the cost of performing frequent preventive replacements with the infrequent, but more costly, corrective replacements. To identify a policy that does this, it is first necessary to define the set of feasible candidate policies. For the current example a simple class of polices is the set of control-limit policies; A replacement is carried out if the condition is found above a threshold value, M, or when the component has failed, which is illustrated in Figure 1.2. We also need to be more specific about how different policies are compared. A common optimality criterion is to choose among the candidate control-limit policies, the one that minimizes the long run average cost per time unit. This raises two more issues, which are at the core of the optimization. First, for a given threshold, how do we calculate the resulting average cost per time unit? Secondly, there is an infinite number of thresholds, M, between "New" and "Failure" in Figure 1.2, so how do we perform the search for the threshold that yields the global minimum cost? For the example we have set up here, Abdel-Hameed (1987) provides an analytical expression for the long-run average cost under a given M, when the condition of the system evolves according to a pure-jump Markov process. A one-dimensional search, e.g., a bisection method, between "New" and "Failure" can then be used to identify the optimal threshold. In addition, the author also proves that under the assumption of periodic inspections, the optimal policy is indeed contained in the class of control-limit policies.

1.2 Thesis scope

The above example is perhaps very simple, but it captures the essence of most maintenance optimization models. In all practical applications there are challenges within all of the above six aspects: **system**, **deterioration**, **information**, **maintenance**, **utility**, and **optimization**. Some challenges are related to modeling, and others are related to optimization once the model is established, but the two things are highly interwoven. This thesis contains work that addresses this interplay of modeling and optimization issues. There is a correspondence between the two since a system model that is too detailed leads to an optimization problem that is intractable, thus heuristics and approximate methods must be employed. On the other hand, if we simplify the system model to facilitate the optimization part, we obviously run the risk of not adequately capturing the dynamics, hence we will be finding a policy that works very well on a system model that does not correspond to reality.

The mathematical framework we use in this thesis to describe the system and its dynamics is the Markov Decision Process (MDP). The MDP framework is very versatile, and maintenance models based on MDPs have been applied successfully in various domains. We provide examples of applications in the next section. The generic example in Section 1.1 is simple enough that it could be stated without using the MDP formalism. For more complex systems, such as systems with multiple deteriorating components, the MDP is a particularly useful modeling tool. Furthermore, the analytical optimization approach that Abdel-Hameed (1987) uses for the Section 1.1 example is infeasible for most multi-component systems, due to their complex dynamics. For MDP models there is, however, a standard toolbox for optimization, namely Dynamic Programming (DP). Provided that the MDP meets a number of fairly weak conditions, a DP algorithm will be able to identify a globally optimal maintenance policy. In this thesis, we focus on the application of MDPs and DP for multi-component systems. We present results regarding the generality of the systems that can be modeled with MDP and the computational limitations of using DP for exact computation of optimal policies.

An assumption in the example in Section 1.1 is that the system deteriorates according to some stochastic process. It is common in theoretical studies of maintenance policies to make such an assumption. However, for a real-world system it is non-trivial to model the deterioration. In fact, predicting the time until failure from monitored system variables is an academic field in itself. The optimization of maintenance activities can therefore be seen as a second step, which follows after a method of predicting system failures from the available information is in place. The focus of this thesis is on the optimization part, and therefore we will assume a specific stochastic process to represent deterioration in a generic system.

1.3 Applications

In this section we provide examples of real-world systems, where maintenance optimization models are applied. The simple example in the previous section was an initial attempt to describe the overall maintenance problem in its most generic form. The decomposition of the problem into the six aspects **system**, **deterioration**, **information**, **maintenance**, **utility**, and **optimization** is similar to that of Dekker (1996), where the author surveys all scientific papers that involve applications of maintenance optimization models up to the year 1996. The author was able to identify 43 case studies where models have been used on real data. The most popular areas are equipment and vehicle replacement, maintenance of electric power stations, and road maintenance.

A successful example of a road maintenance application is presented in Golabi et al. (1982), where the system under consideration is the 7400-mile road network in Arizona. Over time, the wear from vehicles driving on the roads and the constant exposure to the weather lead to cracks and unevenness in the pavement. Resurfacing is therefore needed in order to keep the road condition up to U.S. federal standards. The price of a resurfacing increases with the thickness of the asphalt layer. A constrained MDP model is formulated for each individual mile of pavement in the network, and a policy that outputs the appropriate resurfacing thickness depending on the current road condition is obtained.

The model in Golabi et al. (1982) is a high-level model for the entire road network. A model presented by Medury and Madanat (2013) incorporates more localized details in the utility. These are aspects such as budget constraints and costs for a decreased road capacity or rerouting of traffic when maintenance is performed. The resulting model is much more complex and harder to optimize, but the result is a maintenance policy on a more tactical level.

Similar to roads, bridges have a life span of several decades. In general, infrastructure management is a domain where maintenance optimization models are well suited. The sheer amount of maintainable units and the fact that public funds are involved require a quantitative method for prioritizing maintenance projects. In Papakonstantinou and Shinozuka (2014a), it is reported that the AASHTOWare^{TM} maintenance management system is used for over 750,000 structures in the United States, and it uses MDP as its core optimization procedure. There is still room for improvement of the models used in such management software, and academic work on the subject is ongoing. For instance in Andriotis and Papakonstantinou (2019), a truss bridge with two substructures of 25 components each and a planning horizon of 70 years is considered. The steel trusses are subject to corrosion, and the rate of corrosion increases with the number of years of exposure without maintenance. Furthermore, inspections of the truss components are imperfect, which adds an extra layer of uncertainty. Possible maintenance activities range from minor interventions such as cleaning and repainting, which only delay the aging process, to structural strengthening and replacements that also improve the damage state. Accounting for all this, plus the economic savings from maintaining several of these components si-

multaneously rather than at different times, makes the optimization problem very challenging.

According to Arismendi et al. (2021), the bridge management systems used in Denmark, Norway, Finland, France, South Africa, United Kingdom, China, and South Korea all prescribe inspections from predefined procedures, and a global condition rating is assigned to the structure among a very limited number of discrete levels. The authors consider the specific case of Norway, a country with 18,000 road bridges, where safety regulations dictate the maximum delay for the next repair activity based on the condition found at an inspection. The discrete condition state formalism fits well in the MDP framework that is used in the above mentioned studies. However, the delay between the decision and the action related to maintenance is difficult to incorporate in an MDP model, which is why Arismendi et al. (2021) presents another framework based on a piece-wise deterministic Markov process.

One of the challenges of implementing maintenance optimization models is the specification and estimation of a deterioration model, which require historical condition and failure data. In some cases, this data is being recorded for regulatory safety reasons, however the information may be limited to a high-level rating system such as the bridge examples above. Welte et al. (2006) treats the case of hydro power plants in Norway, where such a rating system is already in place for the main components, which are the generator, the turbine, and the cooling water and drainage equipment. The high-level rating system may not be ideal for modeling the long-term behavior of the components. However, the time, effort, and cost required for gathering more accurate data can be substantial and possibly a bad investment. Unless there is evidence that large savings can be gained from the resulting optimized maintenance policy, a simplified model of the system dynamics can be estimated from the discrete condition states. In Welte et al. (2006), the authors model the deterioration as a Markov chain, and use this as a basis for optimization of the inspection intervals.

The purpose of public roads and bridges is not to create a profit, but to provide a mode of convenience for their users. Therefore, it makes sense to constrain the allowed amount of downtime or decreased capacity due to maintenance work by requiring a minimum level of availability and reliability. Contrary to this, the cost of downtime for a production system such as a wind turbine is easily visible through lost revenue from the output power. Choosing a meaningful utility therefore appears to be easier for production systems. There are, however, other challenging problems for maintenance optimization for offshore wind farms. For instance, seasonal weather conditions pose an additional difficulty, since harsh weather restrains the possibility of reaching the offshore structures. Byon and Ding (2010) proposes a maintenance optimization procedure based on the Partially Observable Markov Decision Process (POMDP) framework that accounts for the possibility that maintenance must be postponed or put on hold if weather conditions become too harsh.

In the above studies, the overall system has been the main focus point. In Elwany et al. (2011), the authors perform a in-depth case study for a specific type of component found in most rotating machinery, namely a rolling bearing. Accelerometer signals are used to measure the vibration magnitude, which increases as the bearing

wears down. A geometric Brownian motion is proposed for modeling the exponential nature of the sensor signal. Under the same cost and inspection assumptions as in the example in Section 1.1, the authors show how the optimal policy is also a control-limit policy. Furthermore, to account for unit-to-unit variability, normally distributed priors are assumed for the parameters in the deterioration model. The optimal control-limit is then adjusted with real-time observations of the vibration signal.

1.4 Time-based & condition-based maintenance

The example in Section 1.1 and all applications mentioned in Section 1.3 assume that the level of deterioration can be obtained by inspecting the system. The decision whether or not to perform maintenance is based on this information. This practice is known as Condition-Based Maintenance (CBM).

Another and more traditional approach, is to base the decision on the elapsed time, which is known as Time-Based Maintenance (TBM). For the example in Section 1.1, suppose that the deterioration of the system cannot be observed, only the time of failures. It is then not possible to explicitly model the deterioration process. Instead the failure rate as a function of the component age would serve as the basis of the optimization. Now, it is still possible to perform preventive replacements, however, the time since the last replacement is the only information we can utilize. The optimal policy in this case is an age threshold, say T, at which we replace the system if it has not failed yet. This is illustrated in Figure 1.3.

One of the assumptions of the initial CBM model was that the system is inspected periodically. Provided the time between inspections is short enough, it is readily seen how the CBM policy is superior to a TBM policy. Due to the randomness of the deterioration process, some realizations will exhibit a slower increase than others. If at one realization, the system is bound for failing before time T, the TBM policy will not prevent this failure, thus we incur the high corrective cost. In the opposite situation, where the system will fail sometime after T, the TBM policy will replace the system too early, which affects the long-run average cost negatively. In both scenarios the CBM policy is more likely to replace the system just before failure occurs, thus avoiding the corrective replacement cost and getting the most out of the system's useful life.

The potential savings from using CBM over TBM have lead to an increased focus on implementing CBM in industry. During the PhD project, we have been in contact with different companies that are currently doing so. Through these exchanges, we have identified a number of common practical challenges. In Chapter 2 we present these challenges and relate them to the generic models that we investigate. The interest in CBM is not limited to industry, but is also reflected in academic work, which in the last few decades has shifted from developing and studying TBM models to CBM models (Alaswad and Xiang 2017; Jonge and Scarf 2020).

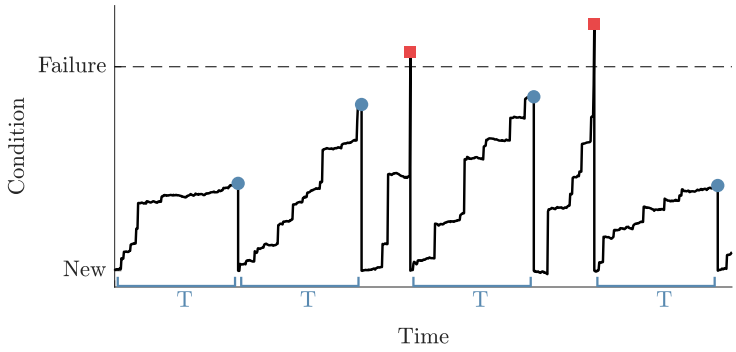


Figure 1.3: A TBM equivalent to the CBM control-limit policy in Figure 1.2. Corrective replacements (red squares) are triggered upon failure, and preventive replacements (blue circles) are performed if the age of the system reaches a threshold age, T.

In general, all technical systems that are expensive and have an important function require maintenance. It is safe to say the most common process for designing a maintenance policy is not mathematical modeling and optimization. Indeed, both the CBM and the TBM approach can be employed without undergoing a rigorous optimization. The most common ways of designing a maintenance policy are through recommendations from the system manufacturer or simply by the best judgment of the maintenance responsible based on his or her experience with the equipment (Ahmad and Kamaruddin 2012).

1.5 The role of maintenance models

There are two purposes of academic papers on maintenance optimization that should be highlighted at this point. First, from the range of applications, summarized in the previous section, it is clear that different systems have different requirements and opportunities that must be taken into account when creating a maintenance strategy. This could for instance be situations, where maintenance must be optimized together with a second objective, or if there are specific practical constraints that go against the common assumptions. For any particular application, the specific circumstances often lead to a tailor-made policy and optimization procedure. A major part of maintenance literature is concerned with developing the necessary mathematical models and algorithms on a case-by-case basis.

The second purpose is to derive general insights about the effectiveness of different maintenance practices under varying assumptions about the system. Some studies consider generic models that mimic a common phenomenon from real-world systems and then perform sensitivity analyses on the system parameters. The goal is to

identify, which parameters that are crucial for the performance of a given policy. The models may be of limited practical use, since there are certain quantities in a real-world system that can be very difficult to estimate, such as the accuracy of the information we observe and the effectiveness of maintenance activities. But, knowing that these aspects are in fact present in real-world systems, valuable insights can still be gained by considering the behavior of a generic model that mimics the real phenomenon.

1.6 Optimal policies

In Section 1.3 we summarized a number of applications, where MDPs have been used as the modeling framework. All of the mentioned examples consider multi-component systems, which is also the category of systems we consider in this thesis.

In multi-component systems, the components are connected in ways that render threshold policies like the CBM policy in Section 1.1 and the TBM policy in Section 1.4 suboptimal. This can for instance be a consequence of dependence in the deterioration processes, the reliability of the overall system, or joint setup costs when initiating maintenance activities.

Identifying an optimal policy in an MDP can be done using DP algorithms. The computational requirements of doing so grow exponentially with the number of dimensions in the MDP state description, which is known as the *curse of dimensionality*. In a multi-component maintenance model it is natural to have at least one state variable per component, in which the relevant information about the component's condition is stored. Therefore, optimal maintenance policies in multi-component systems are hard to obtain.

Most studies that have a practical application in mind resort to approximate solution methodologies, since the number of components is often too large to be handled with exact methods. However, for a moderate number of components, it is still feasible to compute optimal policies. In this thesis, we will focus on computing and analyzing optimal policies for as large systems as possible.

Among the approximate solution approaches is reinforcement learning, which is currently receiving a lot of attention in the academic world, due to a number of very successful applications, see e.g., Silver et al. (2016), Andrychowicz et al. (2020), and Mnih et al. (2015). With the development of these methods it is likely that the application of maintenance optimization models based on MDPs become more widespread. Reinforcement learning algorithms are in general not guaranteed to converge to the globally optimal policy. From the perspective of investigating the behavior of generic multi-component systems through sensitivity analyses, the generality of any given result is compromised somewhat by a possibility that a sub-optimal maintenance policy was used. We avoid this, by only considering optimal policies. We elaborate further on this point in Chapters 2 and 5.

1.7 Outline of the thesis

Given that the complexity of modern technical systems and the performance requirements of them are only increasing, there still remains a lot to be uncovered about how to maintain these systems in an cost-efficient way. The work presented in this thesis falls under the second purpose of maintenance modeling mentioned in Section 1.5. Throughout the thesis, we consider generic models of multi-component systems, and attempt to derive general insights about them.

A number of choices regarding the models and solution methods are constant in all chapters. We use the MDP modeling framework, and we use DP to obtain optimal policies. Components are assumed to deteriorate according to a gamma process. Maintenance actions are limited to replacement of components as in Section 1.1, i.e., the components are brought back to an as-good-as-new state. We assume a fixed set-up cost that is incurred whenever we perform at least one replacement of at least one component. Under these assumptions we consider both the TBM approach and the CBM approach and investigate the relationship between system complexity, model accuracy, computational requirements, and performance of the optimal maintenance policy.

There are no closed-form solutions to the optimization problems in the multi-component models we consider. The results we present are therefore based on numerical experimentation. The rest of the thesis is comprised of 5 chapters, which are summarized below:

- In chapter 2 we review the basic theory of MDPs, its extensions, and DP. We also relate this theory to other modeling and optimization techniques that are prevalent in maintenance literature. The specific assumption about the system we listed above are quite general, and so the review serves as motivation and context for these choices and also for the experiments we conduct in later chapters based on them.

- In Chapter 3 the computational limitations for obtaining optimal policies in multi-component models formulated as MDPs are investigated. The overall objective is to assess the practical limit for the number of components in the system, which we found had not been documented before. Based on the system assumptions listed above, we formulate both a CBM version and a TBM version of the problem. The models themselves are also novel. The combination of MDP with continuous and stochastically dependent deterioration of components has not been considered before. A multi-component TBM model constructed from an underlying unobserved deterioration process is also new.

- Chapter 4 contains a study of the different discretization methods that can be used when combining continuous deterioration processes with DP. The theory of solving general MDPs with uncountable state spaces is based on proofs that the optimal policy can be approximated arbitrarily well with a sufficiently fine

discretized versions of the problem. For the case maintenance optimization models, a few different ways of performing the discretization have been suggested. However, the comparison of the methods has not been carried out before. Our investigation led to some interesting results regarding the DP algorithm performance and methodological aspects of MDP for maintenance optimization.

- In Chapter 5 we compare the performance of optimal TBM and CBM policies in multi-component systems. The models constructed in Chapter 3 provided a unified framework for comparing the performance of the two maintenance approaches. Previous comparisons of TBM and CBM have either been limited to single-component systems, or have only compared heuristic policies. We make the comparison by the performance of the optimal policies given the available information in the respective approaches. The effects of varying setup costs, the number of components, and degree of stochastic dependence are investigated.
- A final discussion of the overall findings throughout the Ph.D. project are given in Chapter 6 along with some directions for further research.

CHAPTER 2

Modeling & Optimization

This chapter is an overview of the literature and theory that is the foundation of the work in the later chapters.

We start the chapter by summarizing the basic theory of MDP and DP, which are the modeling framework and optimization method we use throughout the thesis, respectively. In Section 2.1 we also mention the extensions of the basic models and algorithms as they demonstrate the broader possibilities and limitations of general maintenance optimization.

The models and algorithms used for modeling systems and optimizing maintenance activities come in as many varieties as the real-world systems that require maintenance. In fact, in application papers, problem specific factors often lead to a tailor-made maintenance model (Dekker 1996), which is also true for the examples of applications we presented in Section 1.3. The specific MDP models we consider in this thesis are quite generic, and even though they cover many different phenomena that may be found in real-world systems, they are not all-encompassing. In Section 2.2, we provide an overview of the terminology and concepts in maintenance modeling. The objective is to provide a frame of reference for the contributions of the models and results we present in the later chapters.

Although we focus on DP, there are a number of other popular approaches to maintenance optimization. In Section 2.3, we categorize the different optimization approaches and give a high-level comparison of their different advantages and shortcomings.

Finally in Section 2.4, we also discuss the practical aspects of applying maintenance optimization models in real life. As point of reference, we draw on a case study of railway tamping optimization, and exchanges we have had with a handful of large companies over the course of the PhD project.

2.1 Markov decision processes & dynamic programming

In this section, we provide a summary of the basic theory of MDPs and DP, since the models and solution algorithms we consider all through the thesis belong to these frameworks. We attempt to keep the exposition brief by focusing the specific assumptions that we make in later chapters. However, the general theory is very rich and MDPs come in many variations depending on the generality of the assumptions about the problem. Furthermore, there are many approximate DP algorithms that extend the capabilities in terms of the problem sizes we can handle. In order to discuss the full perspective of the specific models we consider, we also provide a minimal account of some the MDP and DP extensions that go beyond the basic definitions.

Most of the notation we use is adopted from the book by Puterman (2005). In later chapters, we sometimes for convenience deviate slightly from the notation we use below.

2.1.1 MDP definition

An MDP in its simplest form is a controlled Discrete-Time Markov Chain (DTMC), where the transitions between states are influenced by actions chosen by a decision-maker. The goal for the decision-maker is to choose actions that maximize a predetermined performance criterion. The MDP is characterized by the following elements:

- a set of decision epochs, $\{0, 1, \ldots, T\}$, $T \leq \infty$,
- a set of states, S,
- a set of actions, A,
- a reward function, $r : S \times A \to \mathbb{R}$, and
- a transition probability function, $p : S \times S \times A \to [0, 1]$.

The decision epochs, $\{0, 1, \ldots, T\}$ are discrete points in time, where actions are chosen by the decision-maker. The time between two epochs is called a period. If $T < \infty$ then the problem of choosing actions is called a finite-horizon problem. If $T = \infty$ the process continuous forever, and the problem is called an infinite-horizon problem. All problems we consider are of the infinite-horizon type. The MDP can also be defined such that r and p are allowed to change at each epoch, denoted with a subscript r_t and p_t. This is mostly relevant for finite-horizon problems, hence we assume that r and p are time-homogeneous here.

The set of states, S, and the set of actions, A, can be uncountable, countable infinite, or finite. We consider finite state- and action spaces, since the resulting

MDPs can be solved with general-purpose algorithms. Such models are also called finite MDPs. We discuss the case of uncountable S Section 2.1.4.

At any given epoch, the following events take place. The process occupies a state, $s \in S$. The decision-maker chooses an action, $a \in A$, and receives the reward, $r(s, a)$. The process transitions to a new state, $s' \in S$, with probability $p(s'|s, a)$. Because the decision-maker wants to maximize the received rewards, the action, a, could be chosen such that $r(s, a)$ is maximized. However, at the same time the action must be chosen such that we increase the probability of obtaining large rewards in subsequent epochs. That is, a must be chosen such that $p(s'|s, a)$ is large for more "favorable" states, $s' \in S$. This balance between immediate rewards and guiding the process towards maximizing the future rewards is at the heart of sequential decision-making.

With the above definition of the reward function, r, we have assumed that rewards are deterministic, which is true in the maintenance problems we consider in the later chapters. In general, the reward received in a period may also be random, for instance by being dependent on the subsequent state, s'. In this case $r(s, a)$ represents the expectation over the possible rewards, as this quantity is sufficient for choosing optimal actions under the common optimality criteria we define below in Section 2.1.2.

Maintenance example The following is an old example from Derman (1963) of a simple maintenance model that can be formulated as an MDP. An operating system is inspected at equally spaced points in time, $t = 0, 1, \ldots$, and at each inspection it is classified into one of $D + 1$ states, $S = \{0, 1, \ldots, D\}$. The system is considered new (inoperative) if it is in state 0 (D). After each inspection, we can choose to replace the system with a new one ($a = 1$), unless the system is inoperative in which case a replacement is compulsory. Upon replacement, a new system starts in state 0 at the following inspection. If we do not replace the system ($a = 0$), it moves from state i to state j with probability q_{ij}. Hence, the action space is $A = \{0, 1\}$ and the transition probabilities are given by

$$p(s'|s, a) = \begin{cases} q_{ss'} & a = 0, s < D, \\ 1 & s' = 0, s = D \text{ or } s' = 0, a = 1. \end{cases} \tag{2.1}$$

The cost of replacing the system before it becomes inoperative is $c_p < 0$, and the cost of replacing an inoperative system is $c_c < 0$, where $c_c < c_p$. The reward function is therefore given by

$$r(s, a) = \begin{cases} c_p & s < D, a = 1 \\ c_c & s = D \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

2.1.2 Policies and optimality criteria

The mathematical object used to describe the decision-makers preference for choosing an action, $a \in A$, is called a policy, which we denote by $\pi : S \to A$. In all the MDPs we consider, we assume finite S and A and stationary rewards and transition

probabilities. For this class of MDPs it can be proven that the optimal action at any given epoch depends only on the current state of the system. Hence, the optimal policy is contained in the set of mappings from S to A, denoted Π, and we need not consider the more general randomized and history-dependent policies (Puterman 2005, Theorem 6.2.10 and Theorem 8.4.5).

Optimality can be defined in a number of different ways. We consider two different optimality criteria, namely the expected total discounted reward, and the average reward criterion. Supposing we start the MDP in an initial state $s_0 \in S$, a policy, $\pi \in \Pi$, induces a DTMC of visited states, X_t, $t = 0, 1, \ldots$, where $X_0 = s_0$ and $P(X_{t+1} = s' | X_t = s) = p(s'|s, \pi(s))$. Let $\gamma \in [0, 1)$ denote a discount factor. The expected total discounted reward when following policy π and starting in state $s_0 \in S$ is then defined as

$$v_\gamma^\pi(s_0) = E\left[\sum_{t=0}^{\infty} \gamma^t r(X_t, \pi(X_t))\right], \tag{2.3}$$

where $v_\gamma^\pi : S \to \mathbb{R}$ is known as the value function of policy π. Under the expected total discounted reward criterion, a discount-optimal policy $\pi^* \in \Pi$ is one that satisfies

$$v_\gamma^{\pi^*}(s_0) = \max_{\pi \in \Pi}\{v_\gamma^\pi(s_0)\}, \quad \text{for all } s_0 \in S. \tag{2.4}$$

Assuming that the reward function r is bounded, v_γ^π is well-defined. Furthermore, S and A being finite implies that Π is a finite set, hence the maximum is attained in the definition of $v_\gamma^{\pi^*}$.

The average reward when following a policy, $\pi \in \Pi$ and starting in state $s_0 \in S$ is defined by

$$g^\pi(s_0) = \lim_{N \to \infty} \frac{1}{N} E\left[\sum_{t=0}^{N} r(X_t, \pi(X_t))\right], \tag{2.5}$$

and an average-optimal policy $\pi^* \in \Pi$ is one that satisfies

$$g^{\pi^*}(s_0) = \max_{\pi \in \Pi}\{g^\pi(s_0)\}, \quad \text{for all } s_0 \in S. \tag{2.6}$$

The existence of the limit in Equation 2.5 and the average-optimal policy requires more intricate analysis and additional assumptions about the DTMC, X_t, induced by policy $\pi \in \Pi$. Specifically, for any $\pi \in \Pi$, X_t must be unichain, meaning that it contains at most one recurrent class of states, see Puterman (2005, chp. 8). In Chapter 4, where we first consider the average reward criterion, we give an argument that shows how the unichain assumptions holds for the models we consider.

2.1.3 Dynamic programming

Dynamic programming is the name used to collectively describe the algorithms for finding optimal policies in MDPs. We describe the two most common algorithms, Value Itereation (VI) and Policy Itereation (PI), which are algorithms for finite MDPs.

In MDPs with countable infinite or uncountable state spaces, optimal policies can be approximated by solving a finite-state MDP (Puterman 2005; Chow and Tsitsiklis 1991; Rust 1997), so VI and PI are also fundamental tools for solving these problems. Here we summarize VI and PI for the expected total discounted reward criterion, which we use in Chapter 3. Both algorithms exist in a version for the average-reward optimality criterion as well, and we consider these in Chapters 4 and 5.

In the case of the expected total discounted reward criterion, the value function, $v_\gamma^{\pi^*}$, of a discount-optimal policy, π^*, can be shown to be a unique fixed point of the following set of $|S|$ nonlinear equations, known as the Bellman equations,

$$v(s) = \max_{a \in A} \left\{ r(s,a) + \gamma \sum_{j \in S} p(j|s,a) v(j) \right\}, \quad \forall s \in S, \tag{2.7}$$

and with a solution to these equations, we can recover the discount-optimal policy.

Value iteration The VI algorithm solves the system of equations (2.7) iteratively starting at some arbitrary guess, v_0, and calculating a sequence of value function estimates, v_n, $n = 1, 2, \ldots$ by

$$v_{n+1}(s) = \max_{a \in A} \left\{ r(s,a) + \gamma \sum_{j \in S} p(j|s,a) v_n(j) \right\}, \quad \forall s \in S. \tag{2.8}$$

This sequence converges to the fixed point, $v_\gamma^{\pi^*}$, and for a given tolerance, $\epsilon > 0$, we terminate the algorithm when

$$\max_{s \in S} |v_{n+1}(s) - v_n(s)| < \epsilon \frac{1-\gamma}{2\gamma}. \tag{2.9}$$

When this criterion is satisfied, we have $\max_{s \in S} |v_{n+1}(s) - v_\gamma^{\pi^*}(s)| < \epsilon$ and a final policy, π, whose corresponding value function deviates at most ϵ from $v_\gamma^{\pi^*}$, can be recovered by

$$\pi(s) \in \arg\max_{a \in A} \left\{ r(s,a) + \gamma \sum_{j \in S} p(j|s,a) v_{n+1}(j) \right\}, \quad \forall s \in S, \tag{2.10}$$

where $\arg\max_{a \in A}\{\cdot\}$ is the subset of elements in A at which the maximum of the expression in the braces is obtained. Note that we are not guaranteed that the final policy, π, is also a discount-optimal policy as defined in Equation (2.4). However, we can obtain an arbitrarily close approximation to an optimal policy, by choosing ϵ sufficiently small.

Policy iteration The PI algorithm is different from VI in that it computes a sequence of improving policies π_n, $n = 0, 1, 2, \ldots$, starting from an arbitrary policy, $\pi_0 \in \Pi$, and then terminating when the policies cannot be improved anymore. Each iteration in PI consists of two steps. In the first step, known as policy evaluation, the value function, $v_\gamma^{\pi_n}$, corresponding to the current policy, π_n, is calculated. In the second step, policy improvement, a new and improved policy is then computed as

$$\pi_{n+1}(s) \in \arg\max_{a \in A} \left\{ r(s,a) + \gamma \sum_{j \in S} p(j|s,a) v_\gamma^{\pi_n}(j) \right\}, \quad \forall s \in S. \tag{2.11}$$

An informal argument for why π_{n+1} is better than π_n is that the maximizing values in Equation (2.11) correspond to the expected total discounted reward of a policy that chooses optimal actions at the very first decision epoch, and then take actions according to π_n from there on. Therefore π_{n+1} will be no worse than π_n. Note that there can be more than one maximizing action in Equation (2.11), so we choose $\pi_{n+1}(s) = \pi_n(s)$ whenever it is possible. The algorithm terminates when $\pi_{n+1} = \pi_n$, and this happens after a finite number of iterations since the number of policies, $|\Pi|$, is finite.

The policy evaluation step goes as follows. Similar to the value of the discount-optimal policy, the value function, v_γ^π, of any given policy, $\pi \in \Pi$, is a unique fixed point of the following system of linear equations

$$v(s) = r(s, \pi(s)) + \gamma \sum_{j \in S} p(j|s, \pi(s)) v(j), \quad \forall s \in S. \tag{2.12}$$

In vector notation this can be written as

$$(I - \gamma P_\pi) v = r_\pi, \tag{2.13}$$

where $v, r_\pi \in \mathbb{R}^{|S|}$ with elements $v(s)$ and $r(s, \pi(s))$ respectively, and $P_\pi \in \mathbb{R}^{|S| \times |S|}$ with elements $P_\pi = [p(j|s, \pi(s))]_{sj}$ is the transition probability matrix of the DTMC, X_t, induced by π. The linear system of equations can be solved directly by inverting the matrix $(I - \gamma P_\pi)$ in Equation (2.13). This leads to an exact evaluation[1] of the policy π, and as a consequence the final policy obtained in the PI algorithm is discount-optimal. However, if the number of states $|S|$ is very large, as it will be in the MDPs we solve in this thesis, storing the matrix P_π is prohibitive. It is then necessary to solve the linear equations iteratively starting with an initial guess, $\hat{v}_0$, and calculating

$$\hat{v}_{k+1}(s) = r(s, \pi(s)) + \gamma \sum_{j \in S} p(j|s, \pi(s)) \hat{v}_k(j), \quad \forall s \in S, \tag{2.14}$$

iteratively until the criterion in Equation (2.9) is satisfied for some low tolerance, $\epsilon > 0$. Just as for the VI algorithm, the final policy is in this case a close approximation to a discount-optimal policy.

[1] Modulo the inherent rounding error on finite precision digital computers.

Curse of dimensionality In Section 2.1.2 we mentioned that the computational requirements of DP grow exponentially with the number of components in multi-component systems. From the summary of VI and PI above, it becomes apparent why this is the case. Both VI and PI are iterative algorithms, and each iteration involves an update of either the value function or the policy in each state, $s \in S$. The VI update in Equation (2.8) and the policy improvement step in Equation (2.11) perform maximization over the action space and involves an expectation over the subsequent states, hence the number of operations is in the order of $|A||S|^2$. In the example in Section 2.1.1 the system is regarded as a single entity, which can be in any of $|S| = D + 1$ different states and there are only two possible actions, $|A| = 2$. Even if D is very large, the VI and PI algorithms implemented on a modern computer will converge in a very short time. The natural extension of this MDP model to a system consisting of N components, is to have the state of each component belong to the set $\{0, \ldots, D\}$. The state space of the MDP then becomes $S = \{0, \ldots, D\}^N$ where a state $\mathbf{s} = (s_1, \ldots, s_N) \in S$ is a vector containing individual component states. The number of operations in each iteration of VI and PI therefore grows exponentially with the number of components in the system, N.

Besides the standard algorithms, VI and PI, there is a hybrid algorithm, Modified Policy Itereation (PI), and a number of techniques for improving the convergence of the algorithms. These are considered in Chapter 3, where the computational limitations of DP for multi-component maintenance MDP models are investigated.

2.1.4 Extensions

We now proceed with a summary of the possible extensions to the finite MDP and DP theory. Without delving too much into the maintenance terminology (we save that for the next section), we also provide examples from literature that show how these extensions are useful in the context of maintenance.

Uncountable state space The assumption that S is finite is somewhat limiting when modeling maintenance systems. Ideally, we would like to consider models with $S = [0, \infty)$ (or $S = [0, \infty)^N$ for multi-component systems) in order to model gradual deterioration realistically. A more general formulation of the MDP in terms of Borel spaces exists, which allows us to consider MDPs with uncountable state spaces. The Bellman equation equivalent to Equation (2.7) switches sums out with integrals

$$v(s) = \max_{a \in A} \left\{ r(s, a) + \gamma \int_S v(u) p(u|s, a) du \right\}, \tag{2.15}$$

where p is now a probability density function rather than a probability mass function as in the case of discrete S. Ensuring the existence and uniqueness of solutions to Equation (2.15) requires appropriate assumptions of measurability of r and p (Puterman 2005; Bertsekas 2012), and analytical solutions to Equation (2.15) are rare. Rust (1986) provides an example of a replacement problem for durable goods, which can

also be thought of as replacement in a deteriorating system. For this problem, Equation (2.15) was shown to have a closed-form solution and the corresponding optimal policy a control-limit policy.

For more complicated models of multi-component systems, the solution to Equation (2.15) can only be found by discretizing the problem (Özekici 1988; Rust 1997), but discretization can be shown to approach the optimal solution, under additional assumptions of Lipschitz continuity of r and p (Bertsekas 1975; Chow and Tsitsiklis 1991). In this thesis, we consider systems with multiple continuously deteriorating components. However, because they are multi-component systems, we do not delve into the theory of the Borel model, but jump straight to the discrete MDP formulation, which can be solved algorithmically.

Semi-MDP The default assumption in an MDP is that decision epochs are equidistant in time and the MDP is therefore a discrete-time model. A Semi-Markov Decision Process (SMDP) is a continuous-time generalization of the MDP, where the time spent in a state is a random variable, with a distribution that can depend both on the current state and the action taken. Decision epochs are then placed at the moments where the state changes. In Chen and Trivedi (2005), a CBM model is constructed in the SMDP framework, where the discrete deterioration state changes according to a Continuous-Time Markov Chain (CTMC) (a.k.a. a Markov jump process), and the duration of maintenance, the duration of inspections, and the time between inspections are all random variables. Using a transformation of the SMDP into an embedded MDP, the optimal policy can be found with ordinary DP algorithms (Puterman 2005).

Constrained MDP Besides iterative DP algorithms, finding the optimal policy in an MDP can also be formulated as a linear programming problem. Although this method is restricted to smaller problem sizes (Sutton and Barto 2018), it is useful because we can include constraints that ensure the limiting probability of the system being in a specific state is either above or below a predetermined level. Such a models is known as a Constrained Markov Decision Process (CMDP) (Altman 1999). In (Altman 1999) solution methods based on ideas from DP are also presented, but we mention the linear programming approach due to its use in an application in a maintenance context, namely the road resurfacing model proposed in Golabi et al. (1982). The states in the MDP represent the condition of the asphalt. Constraints are added to the linear programming formulation for the maximum limiting probability of a road being in poor condition states, and guaranteeing a minimum limiting probability of being in good condition states. Because the model is applied separately to 7,400 one-mile segments of road, these probabilities also have a second interpretation, namely that the expected proportion of all segments in good and bad states is within the limits defined by the constraint. CMDP is therefore one way of handling multiple conflicting objectives.

Partially observable MDP In the normal MDP, the process will at each decision epoch occupy one of the states in S, and actions are chosen according to this state. But, what if we do not know exactly which of the states in S the system is occupying? A Partially Observable Markov Decision Process (POMDP) is a model for this situation. When the system occupies state $s \in S$ and action $a \in A$ is chosen, the system transitions to state $s' \in S$ with probability $p(s'|s,a)$, but, the decision-maker does not see s', only an observation, $o \in O$ with a probability $p_O(o|s',a)$, where O is a set of possible observation. The decision-maker now has to choose a new action based on which state the system is believed to be in, which is described by a probability distribution b over S. Finding a good policy in a POMDP is therefore harder than in the normal MDP, since the set of probability distributions over S has $|S|-1$ continuous dimensions, and approximate algorithms, such as point-based solvers (Pineau et al. 2003), are therefore used for solving POMDPs. In a maintenance context, the POMDP model is useful for CBM scenarios, where the true condition of the system is not fully revealed. For instance, in Papakonstantinou and Shinozuka (2014b), non-destructive techniques for inspections of steel damage in corroding reinforced concrete structures are assumed to be imperfect.

Approximate DP & reinforcement learning As we described in the end of Section 2.1.3, the standard DP algorithms like value iteration and policy iteration cannot be used when the dimension of the state space is large. However, maintenance optimization in systems with many components can be done approximately using Approximate Dynamic Programming (ADP) and Reinforcement Learning (RL). Both RL and ADP are collections of algorithmic techniques designed to overcome the curse of dimensionality (Powell 2011; Sutton and Barto 2018) and to a large extend RL and ADP are two names for the same thing. The difference between ADP and RL is that ADP is concerned with solving a given (large) MDP by exploiting everything we can from its specific structure, whereas RL focus more on model-free algorithms, and the aspect of the agent (decision-maker) learning from direct interaction with an environment (MDP).

In both ADP and RL, the overall basic principle is to learn how to make good decisions through simulation of the MDP. In many of the approximate algorithms this involves estimates, $\widehat{Q}(s,a)$, of the value of each state-action pair, $(s,a) \in S \times A$, so-called Q-values, $Q(s,a)$. The relation between Q-values and the value function in Equation (2.3) of a given policy π is

$$Q(s,a) = r(s,a) + \gamma \sum_{j \in S} p(j|s,a) v_\gamma^\pi(j). \tag{2.16}$$

One of the simplest and most well-known algorithms is Q-learning. At time t during simulation, the system occupies a state s_t, an action a_t is chosen according to the values $\widehat{Q}(s_t,\cdot)$, we receive a reward r_t, transition to state s_{t+1}, the Q-value is

updated by

$$\widehat{Q}(s_t, a_t) \leftarrow \widehat{Q}(s_t, a_t) + \alpha \left[r_t + \gamma \max_{a \in A} \widehat{Q}(s_{t+1}, a) - \widehat{Q}(s_t, a_t) \right], \tag{2.17}$$

and the simulation clock advances to $t + 1$. There are two immediate benefits of using an ADP/RL algorithm like Q-learning compared to DP. First, we avoid the exhaustive DP iterations, where values in each state are updated before moving on to the next iteration. Secondly, we only update the value of relevant states, i.e., we do not spend time updating states that are unlikely to be visited.

In basic Q-learning, $\widehat{Q}(s, a)$ is a table with one value for each state-action pair. For large MDPs, parametric approximations of the Q-values are needed in order to reduce the size of the problem. One way is to use a linear function on a number of preselected features:

$$\widehat{Q}(s, a) = \sum_{f \in \mathcal{F}} \theta_f \phi_f(s, a). \tag{2.18}$$

Here, $\mathcal{F}$ is a set of features with a size much smaller than the number of state-action pairs, $\phi_f(\cdot, \cdot)$ are basis functions transforming states-action pairs to features, and θ_f are weights that are updated regularly during simulation to improve the Q-value estimates.

In Medury and Madanat (2013), ADP/RL is used in the context of Maintenance, rehabilitation and replacement planning in a road network with capacity constraints. Using the notation from the end of Section 2.1.3, the network has N road segments, and vector states $\mathbf{s} = (s_1, \ldots, s_N)$ and vector actions $\mathbf{a} = (a_1, \ldots, a_N)$ represent the condition and maintenance for all segments, respectively. The authors use linear function approximation, where all basis functions are binary indicators such that Equation (2.18) takes the form

$$\widehat{Q}_t(\mathbf{s}_t, \mathbf{a}_t) = \sum_{f \in \mathcal{F}} \theta_{ft} \phi_f(\mathbf{s}_t, \mathbf{a}_t) = \sum_{i \in N} \theta'_{t, G(i), s_i, a_i}, \tag{2.19}$$

where G is a mapping to a set of groups having a homogeneous response to the capacity constraints in the network. A problem instance with 11 road segments is solved using temporal difference learning with a finite horizon, T, hence the subscript t in Equation (2.19).

Another and more flexible class of function approximators are artificial neural networks. In Andriotis and Papakonstantinou (2019), both the value function and the policy are parametrized using separate networks, which is known as an actor-critic algorithm. The algorithm is used to optimize maintenance for a 25-component steel truss bridge, where inspections are imperfect. Because different degrees of maintenance are included in the model, the number of actions becomes larger than 2.25×10^{15}, but using a parametrized policy is a way to overcome this problem.

An example of an algorithm that exploits problem structure, and should therefore be categorized as ADP rather than RL, is found in Jiang and Powell (2015). The

authors device an ADP algorithm for problems, in which the value function is monotone w.r.t. a multi-dimensional state variable. The state space is equipped with a component-wise inequality, denoted $\preceq$, and monotone then means $s \preceq s' \rightarrow v(s) \leq v(s')$ for states s and s' and value function v. The key difference from other ADP/RL algorithms is that the estimate of the value function is updated with a monotonicity preserving operator. The authors demonstrate how this leads to faster convergence by solving a multi-component CBM problem with and without the special operator.

Another ADP example is found in Xia et al. (2008), where the authors prove for a multi-component TBM problem that the optimal policy satisfies a *shortest-remaining-lifetime-first* rule, which simply means that older components are replaced before younger ones. As a consequence, the action space is reduced from all 2^N combinations for replacing N components to choosing how many of the N components to replace. The authors consider a 30-component numerical example and solves it with temporal difference learning with linear approximation of the value function. The action selection step in this algorithm includes a maximization over all actions (similar to Equation (2.10)), which is not feasible without the reduced action space.

A few ADP/RL algorithms, for instance Q-learning, can be proved to converge to the optimal solution, but the majority of algorithms do not come with such guarantees (Sutton and Barto 2018). The converge proof of Q-learning relies on the property that every state in the MDP is visited infinitely many times, so it is an impractical method for computing optimal policies. In practice, ADP/RL algorithms are often terminated when the performance of the approximate policy appears to have stabilized, which requires a subjective judgment (Powell 2011). Furthermore, because we typically use these algorithms on large problems, the only means of assessing how well they work, is to compare with optimal solutions to small instances of the same problem, or to compare with simple heuristic policies.

2.2 General system modeling

We already touched upon the overall constituents of a maintenance model when we described the generic example in Section 1.1. To recap, these are: **system**, **deterioration**, **information**, **maintenance**, **utility**, and **optimization**. Just as these constituents are reoccurring in all maintenance models, there is a common terminology in the academic literature for specifying the finer details within each of the six aspects. In this section, we review the terminology and common assumptions within the first five aspects, and relate it to the MDP theory above. The final aspect, **optimization**, is treated separately in Section 2.3.

As can be seen from the various applications listed in Section 1.3, the **system** can be any physical entity that could require maintenance. However, from a modeling perspective, there exist only two major types of systems, namely single-component systems and multi-component systems. The former can be seen as a subset of the latter, and therefore all concepts that are relevant for single-component systems keep their relevance for multi-components systems. In this thesis, we only deal with multi-

component systems. Therefore, at the end of this section, we provide a separate treatment of the aspects, which are only relevant for multi-component systems.

In the following we only present a general overview of maintenance modeling, and some subtopics have been left out. More detailed reviews are given by Cho and Parlar (1991), Dekker et al. (1997), Wang (2002), Nicolai and Dekker (2008), Olde Keizer et al. (2017), Alaswad and Xiang (2017), and Jonge and Scarf (2020).

2.2.1 Information

Perhaps the aspect with the most influence on the available optimization approaches is the assumptions we make about the incoming information we have about the system as time progresses. The are two things to consider here. First, the type of information we receive about the state of the system and secondly, the points in time, where the information about the state arrives.

Regarding the former, we have already introduced the two major categories in Section 1.4, namely TBM and CBM. From the references collected throughout the PhD studies, it appears that after the turn of the millennium, the majority of academic works in maintenance have focused on studying CBM. One reason is the development of sensor technology for monitoring equipment deterioration (Alaswad and Xiang 2017; Shin and Jun 2015; Jardine et al. 2006). This has made CBM attractive and also accessible to a larger segment of industry. The previous focus on TBM models thus reflects the nature of the problem that industry was facing, with respect to the available information about the systems that were generally present at the time. This is supported by the review on multi-component models by Dekker et al. (1997). With very few exceptions, this review only contains TBM models where maintenance is triggered based on component age information. We primarily use the term TBM to refer to this kind of policy, but in Section 2.3.1 we give an example of another type of TBM policy, where maintenance is triggered by calendar time.

Information about the state In CBM it is assumed that we observe the condition of the system or its components, but we distinguish between two cases. First, we can assume that we observe the true condition of the system. This is done in the majority of CBM studies (Alaswad and Xiang 2017), however, it is not always possible in practice. Secondly, we can consider that the true condition is partially observable by assuming a distribution on the observed state conditioned on the true underlying state. The POMDP framework described in Section 2.1.4 and used in, e.g., Nguyen et al. (2019) and Papakonstantinou and Shinozuka (2014b) is an example of this approach.

All maintenance models, both TBM and CBM, have some notion of system failure. Although TBM models assume that we have no information about possible intermediate condition states, the failure of the system may reveal itself automatically, which is also called self-announcing failures (Jonge and Scarf 2020). The available information in a TBM model may therefore include whether or not the system is functioning. In

a TBM model, where failures are not self-announcing, maintenance must be planned based only on assumed knowledge about the time-to-failure distribution for the system, since we never actually observe the state of the system. An example of this is the Block-Replacement (BR) policy, where the system is replaced at pre-specified times kT, $k = 1, 2, \ldots$, $T > 0$, which is described in more detail in Section 2.3.1.

Besides the intrinsic deterioration of the system, relevant external environment information (temperature, humidity, season, etc.) may also be included in the model if this influences the rate of deterioration or cost of maintenance (Kurt and Kharoufeh 2010; Byon and Ding 2010; Deloux et al. 2009).

CBM inspection schedules In TBM models, self-announcing failures are the only new information that may arrive as time progresses. The arrival of new information is therefore mostly related to CBM, and the possible schedules for when the condition of the system is inspected. The inspection schedule can either be continuous or discrete. Furthermore, the discrete schedule can be divided into periodic and aperiodic inspection. Continuous monitoring of the system condition may refer to a situation, where the system is monitored in real time using sensors (Castro et al. 2020). Periodic inspections is a common assumption in the literature, both because it may be the only practical schedule, e.g. during an annual production plant shutdown (Alaswad and Xiang 2017), and also because it can make the modeling of the system and the optimization problem more tractable. Other times, periodic inspections are mandatory due to safety regulations (Arismendi et al. 2021). An aperiodic schedule can be beneficial if there is a high cost associated with inspections. For instance, in (Grall et al. 2002) a policy is considered, where the system condition at the current inspection determines the time until the next inspection. Inspections are then less frequent as long as the system condition is good, and more frequent when the system is close to failure.

Whether or not failures are self-announcing is also relevant for CBM with discrete inspection schedules. If failures are self-announcing, they may occur in between two inspections, and an appropriate reaction to the failure can then immediately be decided. However, if a failure is hidden, then the system will remain in the failed state until the next inspection, which can have negative economical and safety-related consequences. Klutke and Yang (2002) provides a CBM example with hidden failures, namely power systems with protective relaying, that operate in stand-by mode until needed. Even in stand-by mode the relaying equipment deteriorates, so it is necessary to physically inspect them every now and then, otherwise failures may be hidden until the equipment is suddenly needed.

Time and epochs Following Noortwijk (2009), a maintenance optimization model can also be classified according to the possible moments for decision epochs. In a continuous-time model, decision epochs can be at any time, $t \in [0, \infty)$. This can either be through a continuous inspection schedule in CBM (Castro et al. 2020), or by epochs triggered by self-announcing failures, which may occur anywhere in $[0, \infty)$.

A model is also continuous in time if, for any $t \in [0, \infty)$, there is a policy in the set of admissible policies that has an epoch at time t. For instance, in the BR policy the decision variable, T, for the time between replacements can be set equal to t.

An MDP, like the example in Section 2.1.1, is a discrete-time model. However, through the SMDP framework described in Section 2.1.4, continuous-time MDP models can also be formulated. As described above, the period length in a CBM model with periodic inspections, is sometimes a decision variable. In Xu et al. (2021), Chen et al. (2015), and Sun et al. (2018) joint optimization of maintenance and inspections is done by varying the inspection period and solving a discrete-time MDP for each considered period length. In that sense, an ordinary MDP can also become a continuous-time model.

2.2.2 Deterioration

Here, we cover some of the models used to describe and predict how the condition of the system evolves over time. We use the term deterioration model to cover all different ways a system can go from being new to failed, although this term is sometimes reserved for gradual and observable deterioration.

Which kind of deterioration model is appropriate depends on the available information about the system condition and the nature of failures.

Lifetime distribution When the information about the system state is limited to whether the system has failed or not, it is natural to only model the time to failure, T. The distribution of T is often given via a failure (hazard) rate function, $h(t) = f(t)/R(t)$, where f is the probability density function of T and R is the reliability (survival) function, $R(t) = P(T > t)$. Choices for the distribution for T are, for instance a Weibull distribution (Dekker et al. 1996), an exponential distribution (Assaf and Shanthikumar 1987), a gamma distribution (Haurie and L'Ecuyer 1982), or a "bathtub" failure rate distribution (Archibald and Dekker 1996; Ahmad and Kamaruddin 2012).

Hard and soft failures In CBM for single-component systems, the most common modeling approach is to represent the deterioration of the system by a univariate stochastic process, $\{X_t\}_{t \in \mathcal{T}}$, where $\mathcal{T}$ is either a discrete set or the nonnegative real numbers, $[0, \infty)$. The system failure is modeled as the first passage time of a limit, $L > 0$, i.e., the time of failure is $T = \inf\{t \in \mathcal{T} : X_t \geq L\}$. In Meeker and Escobar (2014) two types of failures are defined, where the limit, L, is either fixed or random, respectively. For some systems, the failure event is obvious, because the system stops functioning. These failures are named hard failures, and it is in general appropriate to let L be a random variable in this case. If X_t represent some physical variable measured on the system (e.g., vibration magnitude, power consumption, etc.), then failures will in general not correspond to X_t crossing a specific level, instead the value of X_t when the system fails will vary from unit to unit. Soft failures on the other hand

are when the performance of the system decreases gradually as the system deteriorates. In this case, L can be a fixed critical level, at which the system performance has become unacceptable. For manufacturing systems, the performance loss can be an increasing defect product rate, and X_t might even represent this quantity directly (Grall et al. 2002). A fixed L may also be motivated by safety-requirements, where the risk of a hard failure is too large for a system operating above the limit. We elaborate further on this point in Section 2.2.4.

Continuous-state processes There are several choices for the process, $\{X_t\}_{t\in\mathcal{T}}$. Examples of processes where the state space is a continuum include the Wiener process (Sun et al. 2018; Guo et al. 2013; Zhang et al. 2018), the inverse Gaussian process (Chen et al. 2015), fatigue crack-growth models (Ditlevsen and Madsen 1996; Kozin and Bogdanoff 1989; Meeker and Escobar 2014; Morato et al. 2019), the compound Poisson process (Klutke and Yang 2002; Ranjkesh et al. 2019), geometric Brownian motion (Elwany et al. 2011), and the gamma process (Andriotis and Papakonstantinou 2019; Shafiee et al. 2015; Jonge et al. 2017; Noortwijk 2009; Nguyen et al. 2015; Mercier and Castro 2019; Grall et al. 2002).

The gamma process is very often used in numerical examples in maintenance optimization research papers. We also use the gamma process as the standard example in all the later chapters, and therefore we describe this particular process in more detail. A random variable, X, is said to be gamma distributed with shape parameter $\alpha > 0$ and rate parameter $\beta > 0$, denoted $X \sim \text{Gamma}(\alpha, \beta)$, if its probability density function, f_X, can be written

$$f_X(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}, \tag{2.20}$$

where Γ is the gamma function. The stationary (time-homogeneous) gamma process is a special case of a pure-jump increasing Lévy process, and it is characterized by the three properties:

1. $X_0 = 0$ with probability one;
2. $X_t - X_s \sim \text{Gamma}(\alpha(t-s), \beta)$ for all $t > s \geq 0$;
3. X_t has independent increments.

The gamma process is suitable when deterioration is caused by monotonically accumulating damage by a sequence of increments, which is a reasonable assumption for many systems. In contrast, a deterioration process such as the Wiener process assigns non-negligible probability to negative increments, thus allowing for spontaneous improvements in the condition. At time t, the expected value of a gamma process is given by $E[X_t] = \alpha t/\beta$, and the variance is given by $V[X_t] = \alpha t/\beta^2$. When $V[X_t]$ is small compared to $E[X_t]$, the trajectories of $X(t)$ are almost linear. Conversely, if $V[X_t]$ is large compared $E[X_t]$, accumulated damage is caused by a few but large jumps. Figure 2.1 illustrates the two cases plus an intermediate case. In the large

variance case, there is not much benefit in using CBM over TBM, since failures are likely to be caused by a single jump, that occurs at an exponentially distributed time (i.e., according to a Poisson process). The same is true for the small variance case, since the lifetime of the system varies very little (Jonge et al. 2017). In Noortwijk et al. (1995) and Noortwijk et al. (1997) it is shown that the gamma process follows minimal assumptions of non-negative and exchangeable increments, where the order of occurrence is irrelevant (l_1-isotropy). In summation, the gamma process is both very versatile in the types of deterioration behaviors it can model, and the justification for using it as a generic example also has a theoretical basis. More details about the gamma process can be found in Noortwijk (2009) which contains a comprehensive overview including theoretical properties, estimation techniques, and the author points out case studies where the gamma process is used to model deterioration of various systems, such as dikes, steel coatings, steel pressure vessels, auto-mobile brake pads, and more.

Discrete-state models The DTMC (Byon and Ding 2010; Kurt and Kharoufeh 2010; Zhou et al. 2013), the CTMC (Chan and Asgarpoor 2006; Arismendi et al. 2021), and semi-Markov chains (Chen and Trivedi 2005) are the most common discrete-state deterioration models. In fact, in one review of CBM optimization (Alaswad and Xiang 2017) these are the only discrete-state deterioration models mentioned. The same is true for the review on general maintenance optimization, Jonge and Scarf (2020).

There are two immediate reasons for assuming a discrete state space. First, there are systems, where an exact real-valued measure of deterioration cannot be obtained, and instead deterioration is categorized into qualitative levels such as "no deterioration", "mild deterioration", "severe deterioration", etc. (Alaswad and Xiang 2017; Welte et al. 2006; Arismendi et al. 2021). Secondly, the assumption of a DTMC deterioration process allows us to model the system as an MDP and optimize the maintenance policy with DP. A number of studies also form the initial model using one of the continuous-state processes described above, and then perform discretization of the state space in order to facilitate the optimization (Hontelez et al. 1996; Elwany et al. 2011; Chen et al. 2015; Sun et al. 2018; Andriotis and Papakonstantinou 2019; Xu et al. 2021). The discretization approach is used in all the work we present in the following chapters, where generic maintenance models are investigated. However, in Section 2.4 we also describe a railway case study, where we found the discretization of af continuous-state deterioration process to be useful in practice.

Condition monitoring, diagnostics, and prognostics We defined CBM (also known as predictive maintenance) in Section 1.4 as maintenance, where decisions are based on information about the condition of the system. Although this definition is consistent everywhere in the literature on CBM, there is a vast amount of work, where the focus lies on the part of obtaining and analyzing the condition information. Jardine et al. (2006) provides a good overview of the general steps involved in CBM, and the summary we provide here is mainly based on this reference.

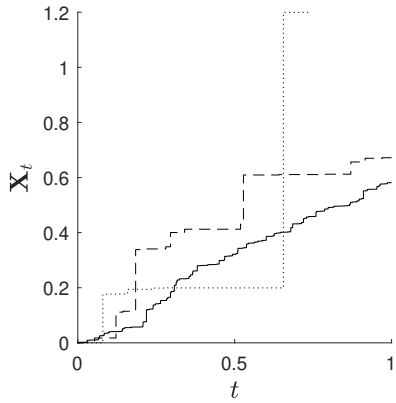


Figure 2.1: Simulated gamma process trajectories with: $E[X_1] = 2$, $V[X_1] = 4.44$ (dotted), $E[X_1] = 1.12$, $V[X_1] = 0.25$ (dashed), $E[X_1] = 1.02$, $V[X_1] = 0.01$ (solid).

The process of collecting measurements related to the health of the system is called condition monitoring. As we have already mentioned in Section 2.2.1, condition monitoring can either be done automatically using sensors mounted on the system or manually by inspection. Examples of data obtained with sensors are vibration data, temperature, acoustic data, oil analysis data, humidity, and pressure. For a better interpretation of the data, a data processing step follows the monitoring step. The method needed for processing depends on the nature of the data. For instance, vibration and acoustic data are waveform type data and require signal processing tools, such as time- or frequency-domain analysis.

After relevant features have been extracted from the data, comes the final step, namely maintenance decision support. From the CBM reviews in Jardine et al. (2006) and Shin and Jun (2015), it appears that this mostly involves diagnostics and prognostics, both of which are terms that describe methods for relating the monitored data to the occurrence of system failures. Simply put, diagnostics are methods for detecting failures when they occur, while prognostics are methods that predict when a failure will happen in the future (also called remaining-useful-life estimation). Various statistical and machine learning approaches for diagnostics are mentioned in Jardine et al. (2006) and Shin and Jun (2015), for instance principal component analysis, logistic regression, cluster analysis, artificial neural networks, support vector machines, and autoregressive-moving-average forecasting. These are all data-driven approaches. For specific system components, physics-based mathematical models are also mentioned as means for diagnostics and prognostics, but stochastic process models such as those described above, are barely mentioned in Jardine et al. (2006) and not mentioned at all in Shin and Jun (2015). In both CBM review papers, prognostics are presented as a method for avoiding critical system failures because maintenance can be performed

preventively before they happen. However, Jardine et al. (2006) also acknowledges that "prognostics, like any other prediction techniques, cannot be 100% sure to predict faults and failures." This leaves us with an optimization problem of balancing the risk of expensive failures with the unnecessary costs of over-maintaining the system.

A stochastic deterioration process, $\{X_t\}_{t\in\mathcal{T}}$, as presented above, describes the evolution of the condition of the system. In the context of studying maintenance optimization of a generic system model, it is common to assume full knowledge about $\{X_t\}_{t\in\mathcal{T}}$, but this is an idealized scenario. In reality, most of what we can measure on the system such as vibrations, temperature, acoustics, etc. are only covariates that indicate upcoming failures. The "true" condition of a system modeled by $\{X_t\}_{t\in\mathcal{T}}$ is therefore an abstraction of the condition monitoring process. Defining a meaningful measure of deterioration and modeling its evolution with any of the above deterioration processes, can be a very difficult task in practice. Nonetheless, in Section 2.4 we argue that a model-based approach is necessary in order to optimize the long-term performance of a maintenance policy.

2.2.3 Maintenance

Just as we need a deterioration model to describe the evolution of the system condition, we also need a model of the impact of maintenance activities. In maintenance literature, there are two main classes of maintenance activities: Corrective Maintenance (CM), which is maintenance as a result of a system failure, and Preventive Maintenance (PM), which is maintenance performed before a failure occurs in order to retain the condition of the system (Wang 2002). Maintenance can also be classified by effectiveness. Perfect maintenance, is when the condition of the system is reset to an as-good-as-new state. This is synonymous with replacement of the system with a new and identical one, in the sense that the condition of the new and the old systems evolve according to i.i.d. stochastic processes. Imperfect maintenance, is any maintenance that reverts the condition of the system to somewhere between the as-good-as-new state and the state right before maintenance was performed (Pham and Wang 1996; Mercier and Castro 2013; Xu et al. 2021). When perfect maintenance is assumed to be the only available action, the system is said to be non-repairable. A light bulb, a brake pad, or a rolling bearing (Elwany et al. 2011) are examples of non-repairable systems. Examples of repairable systems are the truss bridge in Andriotis and Papakonstantinou (2019), where cleaning and repainting of corroded surfaces lowers the deterioration rate of steel truss members, or the different thickness of the pavement resurfacing in Golabi et al. (1982).

Besides timing and effectiveness, the duration of the maintenance activities can also differ. Maintenance planning time, obtaining the necessary spare parts, transporting a maintenance crew to the site, and the maintenance activity itself can all cause a delay between the time of the decision and the time the system is operational again (Jonge et al. 2017; Mercier and Castro 2013). Nonetheless, most studies assume in the model of the system that maintenance activities are instantaneous, which can

be justified if the true duration is negligible compared to the time between failures.

2.2.4 Utility

In the context of optimization, utility is a performance measure used to compare policies, which allows us to search for the optimal policy within a set of admissible policies.

Cost rate The most commonly used utility in maintenance optimization is the average cost per time unit, also known as the cost rate. To give a general definition on a continuous time axis, we use the CBM model in Grall et al. (2002) as an example. We define the cumulative cost up to time t as

$$C(t) = c_i N_i(t) + c_p N_p(t) + c_c N_c(t) + c_d d(t). \tag{2.21}$$

Here, c_i, c_p, c_c and c_d are cost parameters and $N_i(t)$ is the number of inspections in $[0, t]$, $N_p(t)$ is the number of PM actions in $[0, t]$, $N_c(t)$ is the number of CM actions in $[0, t]$, and $d(t)$ is the total amount of time spent in the failed state in $[0, t]$. The cost rate is then defined as

$$\lim_{t\to\infty} \frac{E[C(t)]}{t}. \tag{2.22}$$

Assuming negative cost parameters, i.e., $c_i, c_p, c_c, c_d < 0$, the goal is to choose a policy that maximizes (2.22).

The parameters c_i, c_p, c_c and c_d are mostly self-explanatory, but there are a few remarks to be made. The cost of an inspection, c_i, is only relevant if the inspection schedule is being optimized. The parameter, c_d, is the cost of downtime, which often reflects production losses (Byon and Ding 2010; Dekker et al. 1997; Elwany et al. 2011).

Regarding the PM cost, c_p, and the CM cost, c_c, it is always assumed that $c_c < c_p$ (or $c_c > c_p$ in a formulation with positive parameters). The reasoning is that a failure can induce catastrophic damage to the system and its surroundings and this makes CM actions more complicated and expensive than PM actions. For instance, an engine can be totaled if one of its connecting rods break, so it is cheaper to preventively replace the rod rather than correctively replacing the whole engine (Pham and Wang 1996).

The cost function, $C(t)$, is not always defined as the sum of the four terms included in Equation (2.21). When multiple degrees of maintenance and inspection are possible, multiple cost parameters are assumed, whose values increase with the effectiveness of the action (Golabi et al. 1982; Nguyen et al. 2019). Nonetheless, $C(t)$ is usually comprised of costs that accrue either at a constant rate such as c_d, or lump costs which are accrued at discrete events such as c_i, c_p, c_c. More complicated cost structures are also possible, an example being Broek et al. (2021). Here, production rate and maintenance are jointly optimized, such that a balance is struck between maximizing revenue while minimizing maintenance costs. The authors assume a fixed planning

time for maintenance, and that the deterioration rate of the system is proportional to the controlled production rate. During the maintenance planning time, it can therefore be beneficial to lower the production rate to minimize the risk of failure before the maintenance has been carried out.

Other utilities For manufacturing systems, the cost rate is a suitable utility, but obtaining accurate estimates of the individual cost parameters can be difficult. If the durations of maintenance and inspection are non-negligible, then availability can act as a proxy for cost, as the duration parameters may be easier to obtain than the cost parameters (Klutke and Yang 2002).

Besides using the average cost per time unit, the robustness of a maintenance policy can also be compared through the variability if the incurred costs. In Cherkaoui et al. (2018) this is done by comparing the standard deviation of the cost in a renewal cycle, and in Hong et al. (2014) the authors attempt a similar goal using stochastic dominance rules. These two methods account for the decision-maker's attitude towards risk in the cost rate. The decision-maker's attitude towards the risk of failures, can also be considered by including in $C(t)$ a penalty in form of an operating cost as a function of the deterioration of the system (Medury and Madanat 2013; Andriotis and Papakonstantinou 2019). The specific degree of risk aversion is difficult to interpret in this case. In particular, the operating cost function seems to be chosen rather arbitrarily in the two mentioned references. In Almeida et al. (2015), it is noted that a translation of a system failure into a cost might even be inappropriate for certain systems. For instance, in systems where failures may result in environmental disaster or the loss of human lives, e.g., nuclear power plants, oil distribution systems, or medical equipment, multi-objective optimization methods such as Pareto front solutions and multi-attribute utility theory are more suitable. By using multi-objective optimization, cost minimization is not considered in isolation, but safety and reliability, which are obviously also influenced by the maintenance policy, are taken into account. These methods are reviewed in Almeida et al. (2015).

Although it is difficult to merge safety and cost into a single objective function in service-oriented systems, such as transportation systems, there are cases, where stipulated safety requirements determine the set of admissible maintenance policies. The road network CMDP model of Golabi et al. (1982) described Section 2.1.4 is an example of this. In the multi-component TBM model presented in Xia et al. (2008), safety-critical components in an airplane must be replaced before the components reach a predetermined maximum age. Components older than the maximum age are assumed to have a too high failure rate. In CBM models, a fixed failure level in a stochastic deterioration process can also represent a predetermined safety constraint. We give an example of such a system in Section 2.4.

2.2.5 Multi-component systems

Every modeling aspect we have covered so far are relevant for maintenance modeling of single-component systems. If a system consists of multiple integral components that interact with each other, the modeling and optimization become more complicated. Here we describe the overall kinds of dependences between components, and how they can be accounted for in the modeling.

Whether a system is single or multi-component is a matter of perspective. Any real-world system has a hierarchical structure, because a system consist of sub-systems, and each sub-system again consists of smaller subsystems, etc. (Jonge and Scarf 2020). For instance, a wind turbine is a system, which consists of a rotor, a tower, and a drive train. The rotor is comprised of multiple blades, the drive train consists of bearings, a gearbox, and a generator, and the tower consists of several pieces and a foundation. We can also consider the entire wind turbine as a single component that is part of a larger multi-component system, namely a wind farm. Such a system is subject to the same kinds of dependences, and therefore from a modeling perspective, multi-component is synonymous with multi-unit, multi-item and multi-asset.

Whether we choose to model a system as a single component or as multiple components also affects if the system is non-repairable or repairable. A brake pad in a car can be considered a single-component non-repairable system. However, replacing a brake pad is also an imperfect maintenance action, because the car as a whole is not restored to the as-good-as-new state. From this example we also see that a general feature of a multi-component maintenance model is that at least some of the available maintenance actions only applies to individual components in the system (Jonge and Scarf 2020).

Reviews of multi-component maintenance models, for instance Cho and Parlar (1991), Dekker et al. (1997), Nicolai and Dekker (2008), and Olde Keizer et al. (2017), distinguish between three major types of dependences between components, namely structural dependence, economic dependence, and stochastic dependence.

Structural dependence According to Olde Keizer et al. (2017), there are two types of structural dependence. The first kind is technical dependence, which is present if some components structurally form a part, such that maintenance of one component requires dismantling or maintaining other components as well. The second kind is performance dependence, which concerns the reliability structure of the system, that is, how the failures of individual components affect the overall system performance. The most common reliability structures are: series system, where one component failure causes a system failure; parallel system, where the system retains some performance as long as one component is still functioning; series-parallel system, which is blocks of parallel components in a series; and K-out-of-N system, where the system functions as long as there are at least K functioning components. A K-out-of-N system generalizes series and parallel systems, because a 1-out-of-N system is a parallel system, and a N-out-of-N system is a series system. References to studies

that consider each of these structures and also more complex structures are given in Olde Keizer et al. (2017).

Economic dependence Economic dependence concerns the impact of performing maintenance on multiple components simultaneously. The effects of this can be either negative or positive. Negative economic dependence occurs if simultaneous downtime of components is undesirable, in which case maintenance should be spread out as much as possible (Dekker et al. 1997). Positive economic dependence is when maintaining multiple components simultaneously is cheaper than maintaining the individual components at different times. It is also known by the name opportunistic maintenance, which is named from the situation where an unexpected system shutdown, for instance due to a failure in one component, is an opportunity to maintain other components if these are also worn but still functional. Positive economic dependence is usually built into the optimization problem as a joint setup cost that is only paid once even if multiple components are being maintained. The setup cost can represent the commission when hiring a maintenance crew, the price of putting up a scaffolding around a structure, or the transportation costs of getting a crew to a remote location (Dekker et al. 1996). Positive economic dependence can also be reflected in the reduced system downtime costs, when maintenance durations are assumed to be non-negligible.

Stochastic dependence Stochastic dependence is the dependence between the deterioration processes of each component in the system. This can either be intrinsic or caused by external factors.

Intrinsic stochastic dependence happens, for instance, if the failure of one component sends a transient shock through the entire system, thereby damaging other components (Zhang et al. 2020). Another kind of intrinsic dependence is found in Olde Keizer et al. (2018), which is a load-sharing deterioration model in a parallel system, where the amount of stress on the system is distributed out on all functioning components. In this discrete-time discrete-space model, the number of deterioration increments are Poisson distributed, and when one component fails, the Poisson distribution parameter of the other components increase. Bian and Gebraeel (2014) considers a slightly more complex version of load-sharing. Here, intrinsic stochastic dependence is modeled by each component following a Wiener process with drift, and where an increase in the deterioration level of one component increases the drift parameter of other components.

Stochastic dependence can also arise, if deterioration of components is caused by external forces from the operating environment. For instance, stormy weather can cause damage to the rotor blades on multiple wind turbines in a farm simultaneously (Olde Keizer et al. 2017). In Feng et al. (2015), this kind of dependence is modeled by a Poisson process for the arrival of shocks that affect all components, where the shock damage magnitudes to individual components are independent random variables.

Another way to model stochastic dependence caused by external factors, is to use copula functions (Shi et al. 2020; Hong et al. 2014; Li et al. 2016; Xu et al. 2021; Jiang et al. 2021), which introduce dependence between the deterioration increments of individual components. In Chapters 3 and 5, we consider a multivariate Lévy process, $\mathbf{X}_t = (X_t^1, \ldots, X_t^N)$, where each marginal process X_t^i, $i = 1, \ldots, N$, is a gamma process that represents the deterioration of component i. The gamma process is a Lévy process with intensity measure $\nu_i(x) = \alpha_i x^{-1} \exp(-\beta_i x)$, where α_i and β_i are the shape and rate parameter of component i. The Lévy measure ν of the multivariate process $\mathbf{X}_t$ is represented through a Lévy copula function $C : [0, \infty)^N \to [0, \infty)$. This function links the tail integral $U : [0, \infty)^N \to [0, \infty]$ of the Lévy measure ν given by

$$U(x_1, \ldots, x_N) = \nu([x_1, \infty) \times \cdots \times [x_N, \infty)) \tag{2.23}$$

to the tail integral $U_i : [0, \infty) \to [0, \infty]$ of ν_i by the relation

$$U(x_1, \ldots, x_N) = C(U_1(x_1), \ldots, U_N(x_N)). \tag{2.24}$$

In Chapters 3 and 5, we use the positive Clayton-Lévy copula function

$$C(x_1, \ldots, x_N) = (x_1^{-\theta} + \cdots + x_N^{-\theta})^{-1/\theta}, \quad \theta > 0. \tag{2.25}$$

The parameter θ dictates the dependence of jump sizes, where larger values increase the tendency to observe simultaneous large jumps in each component. Figure 2.2 shows simulated realizations of $\mathbf{X}_1$ with two components for different values of θ. Detailed information about Lévy copula functions can be found in Grothe and Hofert (2015) and Kallsen and Tankov (2006).

2.3 Optimization methods

Using the classification from Nicolai and Dekker (2008), maintenance optimization methods can be classified into exact, heuristic, and policy optimization. The authors define the former as methods designed to find the exact optimum solution to a given optimization problem. An MDP model together with an optimality criterion (e.g. the expected total discounted reward) define an optimization problem, and the optimal solution can be found with DP. Another example of a model formulation, for which there exist exact optimization methods, is Mixed Integer Linear Programming (MILP). We give an example of such a model in Section 2.3.2. A heuristic optimization method, is any method that can find a solution of reasonable quality in a reasonable amount of time. The ADP/RL algorithms for MDPs described in Section 2.1.4 are in that sense heuristic methods. In policy optimization, the search for a solution is restricted to a predetermined class of policies. Usually, the class is specified by a relatively small number of parameters, which then need to be tuned to the best configuration possible.

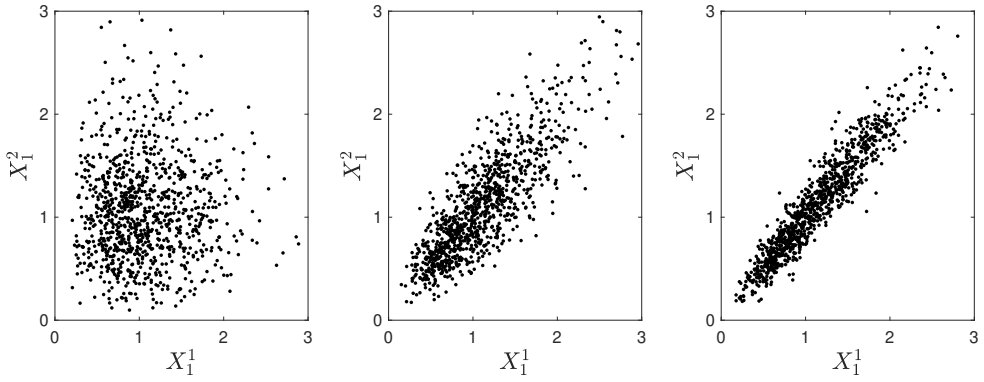


Figure 2.2: Simulated bivariate-gamma process increments, $\mathbf{X}_1 = (X_1^1, X_1^2)$ with $\alpha_1 = \alpha_2 = 5$, $\beta_1 = \beta_2 = 4.45$, and Clayton-Lévy copula depedence with: $\theta = 0.2$ (left), $\theta = 1.5$ (middle),and $\theta = 3.0$ (right).

2.3.1 Policy optimization

The policy optimization approach is perhaps the most common type of optimization in maintenance literature. As mentioned in the survey, Wang (2002), thousands of models for optimal maintenance have been developed, and this survey classifies them according to the design of the maintenance policy. We will not repeat all the classes here, but simply mention the most traditional policies, a few more complicated policies, and the methods used to optimize them.

The basic policy designs for single-component systems are:

Age-Replacement (AR) The system is replaced at age T or failure, whichever comes first. This is the TBM policy illustrated in Figure 1.3. For repairable systems this can be extended by allowing for imperfect maintenance upon failures (Wang 2002).

Block-Replacement (BR) The system is replaced at times kT, $T > 0$, $k = 1, \ldots,$ regardless of the system state at the time of replacements. This is a TBM policy primarily for systems without self-announcing failures, hence when the system fails it remains so until the next planned replacement.

Periodic Inspection and Replacement (PIR) The system is inspected periodically with period Δ and is replaced if the deterioration of the system exceeds a threshold value, M. This is the basic CBM policy illustrated in Figure 1.2.

The above policies are all specified with either one or two parameters. Optimizing for the best parameter configuration requires a method for evaluating the utility (e.g., the cost rate) for a given input of parameters, and a search method over the

set of possible parameter values. For the evaluation of the cost rate, such as it is defined in (2.22) analytical expressions are usually derived using regenerative and semi-regenerative properties of the evolution of the system state and standard renewal theory arguments (Barlow and Proschan 1996; Abdel-Hameed 1987; Grall et al. 2002; Zhang et al. 2020; Guo et al. 2013; Zhou et al. 2013; Castro et al. 2020; Castanier et al. 2005; Dieulle et al. 2002). The expressions usually contain complicated nested integrals that require numerical methods in order to be evaluated. Alternatively, the cost rate evaluation can be performed with Monte-Carlo simulation (Hong et al. 2014; Jonge et al. 2017; Shafiee et al. 2015; Nguyen et al. 2015). If there are only a few parameters to be optimized, an exhaustive search on a fine grid (Jonge et al. 2017; Guo et al. 2013; Nguyen et al. 2015; Castro et al. 2020) or an iterative grid search (Shafiee et al. 2015; Hao et al. 2020) suffices.

The policies developed for multi-component systems are very often based on multiple threshold parameters per component (Olde Keizer et al. 2017). A number of studies of multi-component systems with economic dependence consider variations of the (m_i, M_i)-policy. This policy can either be an extension of the AR policy in the TBM case or the PIR policy in the CBM case. Each component, i, is correctively replaced upon failure, or preventively replaced if its deterioration (or age) exceeds the M_i threshold. Furthermore, if another component $j \neq i$ is being replaced, component i is also preventively replaced if it exceeds the m_i threshold. The structure of this policy is illustrated in Figure 2.3. In van der Duyn Schouten and Vanneste (1990), the AR version is used, where m_i and M_i are age thresholds, and the authors develop an efficient algorithm for the two-component system based on the PI algorithm. In Castanier et al. (2005) the PIR version of the (m_i, M_i) policy is extended in a two-component system with additional thresholds that decide the time until the next inspection depending on the current state. The policy is extended even further in Zhou et al. (2013) to N-component series-parallel systems. When the number of components is large, an exhaustive search through all combinations of possible threshold values is infeasible. Therefore, a tailored heuristic is developed for the task in Zhou et al. (2013), and other studies optimize threshold values using general heuristic search methods such as genetic algorithms (Marseguerra et al. 2002) or ant colony optimization (Liu and Huang 2010).

The fact that the maintenance optimization survey in Wang (2002) categorizes papers according to the type of policy being studied, is itself an indication that policy optimization is used more often than exact methods. In this thesis, the focus is on DP, which is an exact method that is limited by high computational requirements. Tuning a parametrized policy to its optimal configuration can also be computationally demanding, if the number of parameters is large. However, we can stop the procedure at any point and obtain a solution, only it will be a suboptimal configuration of the parametrized policy. The same is not true in a DP algorithm, where completing even a single iteration is sometimes impossible if the state space is very large. Furthermore, the optimal policy may have a complex structure, and therefore be difficult to implement in practice. Although a parametrized policy may lead to a suboptimal solution in terms of utility, it is possibly easier to implement, especially if it has been

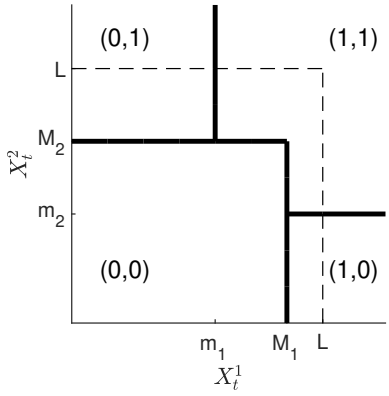


Figure 2.3: Example of an (m_i, M_i) policy in a two-component system. X_t^1 and X_t^2 denote component deterioration and L is a fixed failure limit. Parentheses indicate actions.

tailored to the case-specific circumstances.

2.3.2 Non-stationary policies

So far, we have mainly described optimization of maintenance on an infinite horizon, where the dynamics of the system and the policy obtained from the optimization are stationary. In the following chapters we only consider infinite-horizon models, but there are two alternatives, namely finite-horizon and rolling-horizon models, which have some advantages and disadvantages compared to infinite-horizon models.

In finite-horizon models, the system is only considered up to some final time, $T < \infty$. Finite horizon models can be formulated as MDPs, but MILP formulations are also common. In Caetano and Teixeira (2015), an MILP model is developed for scheduling replacement activities for a 10-year period in a 336km railway track multi-component system. The optimization is performed with a commercial solver, but other than that the authors do not mention if and how the optimal solution is found. The finite horizon formulation adds flexibility to the model, because we can incorporate non-stationary behaviors of the operating environment or cost parameters, and available actions. For instance, in Caetano and Teixeira (2015), the authors include time-dependent budget constraints and an operational constraint that ensures each track segment is replaced exactly once during the planning horizon. However, a finite-horizon model also requires that we specify the scrap value of the system, $r_T(s)$ (using MDP notation), which is the reward received from ending the process in state $s \in S$. Unless the system is in fact taken out of operation at time T, specifying a meaningful $r_T(s)$ is difficult. In Caetano and Teixeira (2015), the scrap value is defined somewhat arbitrarily from a loss of life for not reaching the maximum allowed

number of maintenance operations in the last epoch. Therefore, even though no real system will remain in operation for eternity, the alternative finite-horizon models also have a conceptual shortcoming due to the difficulty of choosing a meaningful scrap value. Sedghi et al. (2021), Nicolai and Dekker (2008), and Moghaddam and Usher (2011) contain more references to MILP maintenance models and the exact- and heuristic methods for solving them.

A rolling-horizon model attempts to get the best from both infinite-horizon and finite-horizon models. Decisions are based on a tentative infinite-horizon plan and are then adapted in each epoch by solving a finite-horizon problem that includes short-term information (Dekker et al. 1997). For multi-component systems it often involves a two-step optimization that first optimizes a policy for each individual component, and then decides in the second step how to group maintenance activities based on a finite-horizon binary mathematical programming formulation, i.e., a set-partitioning problem. In the first step, the long-term policies are found by solving a single-component model for each component in the multi-component system, which is computationally cheap. The second step is a large combinatorial problem and it is therefore often solved approximately using e.g. ant-colony optimization or genetic algorithms (Liu et al. 2018; Bouvard et al. 2011; Vu et al. 2014). Even when the second step is solved exactly, the overall procedure is not guaranteed to be globally optimal because in the second step the optimizer only looks for optimal groupings in the near future and do not account for groupings after the finite horizon (Dekker et al. 1996).

2.3.3 Optimality and assumptions

A simple scheme of the high-level process of maintenance optimization is

1.Real-world system
↓
2.Mathematical model
↓
3.Optimization
↓
4.Maintenance policy

The stages are: Starting from a real-world system (1.), we construct a mathematical model (2.) that mimics reality. Based on the model, an appropriate optimization procedure is chosen (3.), which yields an optimized maintenance policy (4.). Along these stages we have to make some assumptions, but the reason behind an assumption depends on which stage it belongs to. Assumptions at stage 1 are general assumptions that reflect the true nature of the real-world system. Assumptions at stage 2 simplify reality in order to construct the mathematical model, for instance because

the real-world dynamics of the deterioration and failure mechanisms cannot be modeled perfectly. Finally, there are also assumptions at stage 3, which can be additional simplifications that facilitate the optimization.

The general MDP and MILP frameworks do not impose any structure on set of admissible policies, and when an exact method is used for optimization, we can regard the resulting policy as globally optimal. Policy optimization imposes some structure on the set of admissible policies. Between system assumptions (1.) and optimization assumptions (3.), the definition of a globally optimal policy can be different depending on the situation. The question is whether we count assumptions about the structure of a maintenance policy as something that is inherent to the problem, e.g., a practical constraint, or something we impose to make the optimization tractable. This is not always clearly stated in studies that use policy optimization, but in general we regard the policy optimization approach to be a kind of heuristic. That is, unless it can be proven that the globally optimal policy has a specific parametric structure, e.g., a control-limit structure (Abdel-Hameed 1987; Kurt and Kharoufeh 2010; Derman 1963).

2.3.4 How good are the heuristics?

In the previous sections, we have reviewed the different methods of optimization. In this section, we present a sample from the literature, of how well policies obtained from the different approaches compare. In particular, the optimal policies found in MDP models of multi-component systems have on different occasions been compared to simpler, but easier to optimize, parametrized policies. The optimal policies in MDPs are more difficult to obtain, but how much better are they in terms of the resulting cost rate? Also, the optimal policy may have a complex structure and therefore be difficult to implement in practice, as shown in Figure 2.4. On the other hand, optimizing a parameterized policy also requires a model of the system dynamics, so we may as well attempt to obtain the best policy possible.

The (m_i, M_i) policy described in Section 2.3.1 has on multiple occasions been compared to optimal policies. In Haurie and L'Ecuyer (1982), the authors consider a 6-component series system and compare an optimal TBM policy to a special case of the AR version of the (m_i, M_i) policy[2]. In the worst case the (m_i, M_i) policy is only an 1.3% increase in costs compared to the optimal policy. van der Duyn Schouten and Vanneste (1990) is an extensive analysis of a two-component series system where the optimal TBM policy and the AR version of the (m_i, M_i) are compared again. 45 parameter configurations are tested, and in all cases the (m_i, M_i) is less than 0.6% worse than the optimal policy. In Olde Keizer et al. (2016), the authors consider various parameter configurations with $N \in 2, 3, 4$ in a K-out-of-N system and compare an optimal CBM policy to the PIR version of the (m_i, M_i) policy. The increase in cost here lies between 1.5% and 10.2%.

[2]It is proven that if no components have failed it is optimal to do nothing, so only (m_i, D) policies need to be considered, where D denotes the failed state of a component.

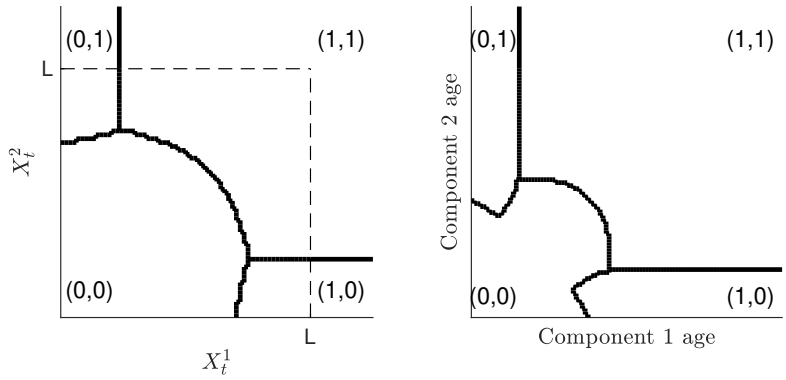


Figure 2.4: Examples of a optimal CBM policy (left) and a optimal TBM policy (right) the a two-component systems from Chapter 5. Parentheses denote actions.

For systems with many components, optimal policies cannot be computed exactly, so policies obtained from approximate ADP/RL algorithms are compared to simpler heuristic policies. For instance, in Andriotis and Papakonstantinou (2019) summarized in Section 1.3, a policy obtained with RL is 5.8% better than an optimized PIR policy.

Some studies also compare optimal policies found with DP to parametrized policies simpler than the (m_i, M_i) policy. In Barreto et al. (2014), a generalization that includes stochastic dependence in the multi-component TBM model from van der Duyn Schouten and Vanneste (1990) and Haurie and L'Ecuyer (1982) is formulated. The optimal TBM policy is here 24.7% better compared to an AR policy that uses the same age threshold for all components. In Olde Keizer et al. (2016), the authors also compare the optimal CBM policy with PIR, AR, and BR policies resulting in cost increases in the ranges 3.8%-20.0%, 40.3%-62.0%, and 21.9%-64.2% respectively. In Olde Keizer et al. (2018) the same model is extended to include stochastic dependence and the optimal CBM policy is again compared to the PIR policy with cost increases between 5% and 52% in the considered examples.

Although the numbers we have presented in this section are not a comprehensive comparison of the optimization methods, they illustrate that there are both situations where a simple policy performs close to optimally in a complex systems and also situations where major savings can be obtained if the optimal policy can be identified.

2.4 Practical aspects

Several hurdles for applying maintenance optimization models in the real world have already been identified in the academic literature (Dekker 1996), and still a large gap

remains between the theory of maintenance and what is actually practiced in industry (Veldman et al. 2011; Fraser et al. 2015). Over the course of the PhD studies, we have been in contact with companies that are currently taking steps towards improving their maintenance processes. Common for all of them is a focus on implementing CBM. The systems in question are, shipping containers and dockside cranes, plastic injection molds, a steel coil production line, power plant components, and railway tracks. From our exchanges with these companies, we acquired a few insights that help to explain the major difficulties in implementing the modeling and optimization techniques presented in the previous sections.

We start by describing the railway track case. Railway maintenance is an application domain, where practical use of maintenance optimization seems realistic, due to a number of features that are inherent to the system. Furthermore, this case is also motivation for some of the assumptions we make for the generic models considered in the later chapters.

2.4.1 Railway case study

In a railway track, the two parallel rails are fastened to perpendicular concrete elements called sleepers, which rest on a bed of crushed stone called ballast. As the track is subjected to the stresses of passing trains, the ballast begins to settle, which may cause misalignments in the geometry of the rails. Ultimately, high levels of misalignment lead to an increased risk of train derailment. To counteract this, maintenance in the form of ballast tamping must be performed, which is done by a machine that lifts the rails and sleepers back in place and packs the ballast underneath. The need for tamping is decided based on rail geometry data, which is collected at frequent measuring campaigns by a track recording car. A dataset from a line between two Swedish cities was provided by *Trafikverket*, and more details about the following can be found in Eegholm-Larsen and Olesen (2021).

The multivariate geometry data is aggregated into summary statistics for each 200 meter segment. A particular statistic, the standard deviation of the longitudinal level, is used as the variable that indicate the need for tamping (Caetano and Teixeira 2015). The European standard 13848-5 (2017) specifies safety limits for this variable. If the limit is exceeded, the entire line must be closed temporarily or the train speeds must be reduced, until the track has been realigned. From a modeling perspective, the standardized limits simplifies the problem as a component failure is then predefined. Because the deterioration rate of this variable is not the same for different segments, tamping is usually not performed on all segments at the same time. The railway can therefore be considered as a multi-component system with a series reliability structure, where components correspond to individual segments.

An example of the data for a single segment is show in Figure 2.5. Because the time between tampings for a segment is usually in the order of years, and because there is some required planning time when scheduling tampings, a discrete-time model formulation with one year between epochs is appropriate. However, there are usually

several measuring campaigns each year, and these are not equidistant, hence the deterioration is more naturally modeled on a continuous time-scale. A uncountable state space is also natural, since the measured geometry variable is continuous. In the analysis in Eegholm-Larsen and Olesen (2021), different candidate models are tested and the one that describes the deterioration best is a Wiener process with drift and error terms. The heterogeneous behavior of segments is modeled with normally distributed drift and infinitesimal variance parameters in the Wiener process. A single-component version of the problem is modeled as an MDP. As a consequence of the doubly stochastic deterioration process, the optimal policy found with DP is a time-inhomogeneous control-limit policy, where the tamping threshold increase with the time since the last tamping. The same form of policy is proven to be optimal in Elwany et al. (2011) for a similar CBM problem for bearings. The complex structure in the policy is not easy to parameterize, and this is only accentuated in the multi-component problem, which justifies formulating the problem in the general framework of MDP.

Discretization is, however, a necessary step for solving an MDP with a uncountable state space as we discussed in Section 2.1.4. This effectively turns the continuous-time continuous-space Wiener deterioration process into a DTMC. In Chapters 3 and 5 we analyze the approach of combining continuous deterioration processes with MDP and DP optimization. Another approach would be to model the deterioration directly as a DTMC, however as it is pointed out in Hontelez et al. (1996), it is very difficult in practice to obtain enough data to fill a one-step transition probability matrix. Indeed, this was also an issue when attempting to fit a DTMC on the geometry variable data. On the other hand, estimating a few parameters of a continuous process is more practical, and this also has the benefit of retaining the intrinsic continuous nature of the system (Grall et al. 2002).

2.4.2 Data

Compared to the other cases mentioned in the introduction of this section, maintenance modeling and optimization seems particularly suited for the railway case. The main reason for this is the mandatory collection of condition data for the tracks, defined according to an international standard. The dataset for the railway case spans more than a decade, but, for some of the other companies we talked to, condition monitoring procedures were at a very early stage, and a proper indicator for deterioration had yet to be identified. Obviously, the condition of the system must be observed for a considerable amount of time, before an appropriate deterioration model can be established, or the remaining useful life can be predicted with accuracy.

In Dekker (1996) they highlight replacements in vehicle fleets, such as buses and trucks, as a fruitful area for maintenance optimization, because a large number of copies of the same system allows for data pooling. The same can be done in the railway case, because all segments are structurally identical. This is not possible for specialized equipment found in a steel coil production line, or power plants. Another

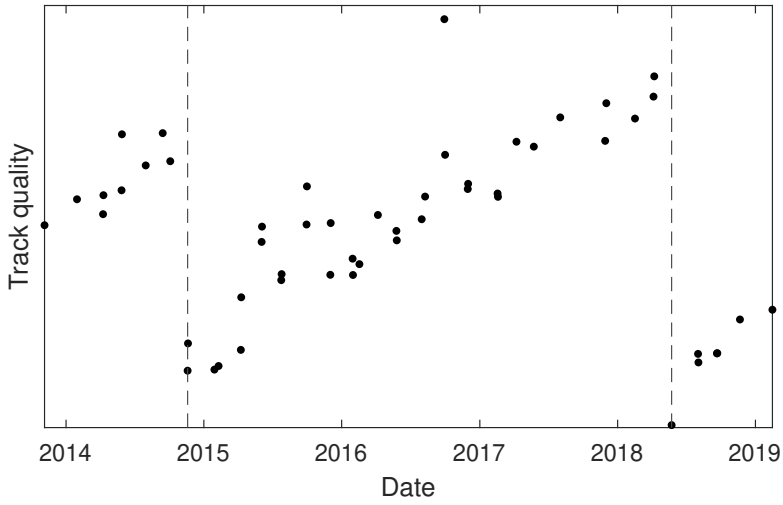


Figure 2.5: Example of geometry measurements over time for a single track segment. Tampings indicates with dashed lines..

reason the railway case is suited for optimization is that the system essentially remains the same for a long period of years. In production systems, new technologies are continuously developed, so instead of replacing components by identical ones, they are sometimes replaced by a newer generation. As a consequence, historical data records become obsolete.

A problem present in most of the cases, was that very few or no failures had ever been recorded. This can be attributed to the historical data records being fairly short, but it can also be a symptom of maintenance being done too often. No recorded failures makes deterioration modeling and failure prediction difficult, since the data contains too little variance to be useful for these purposes (Kulahci et al. 2020). Furthermore, it was sometimes the case that more emphasis was put on collecting and analyzing condition monitoring data, wheres as event data, such as when and how the system was maintained, and cost data would receive little or no attention. In Jardine et al. (2006), the authors also make this observation, and point out that condition monitoring data and event data are equally important.

2.4.3 Prognostics and horizons

Condition monitoring and data-driven prognostics were the main concerns for some of the companies that we have been in contact with. Indeed, condition monitoring is a prerequisite for CBM, and no optimization can be done without a solid condition-monitoring method in place. Prognostics based on data-driven methods are sufficient if we only care about the time until failure and the primary goal is to avoid unex-

pected breakdowns and expensive CM by performing last-minute maintenance activities. However, failure predictions mainly serve as an alarm mechanism, and are not very useful for optimization of the maintenance policy over a longer horizon.

For long-term optimization of maintenance, we argue that a stochastic deterioration model is necessary. At any given time, the decision-maker must decide when maintenance should be performed, given the current information about the system. This requires comparing the different alternatives, but in order to compare the choice of performing maintenance sooner rather than later, we must have a way of accounting for what happens after the maintenance intervention. The stochastic deterioration model provides a way of doing this, since it models the evolution of the system state after maintenance has been performed. Without a model of the system, we are faced with the problem that we do not have any way of weighing the downside of performing maintenance too early, since we do not account for anything that happens post maintenance and therefore do not know how much useful life we sacrifice.

CHAPTER 3

Computational aspects of DP

The following paper is a numerical study of the problem size constraints we encounter when using DP for maintenance optimization of multi-component systems. From a practical point of view it is of course relevant to know the maximum number of components a system can have, where we can still solve the problem with DP in a reasonable amount of time. The motivation for this study comes from the fact that numerous papers model multi-component systems with MDPs and solve them with DP, but only few address the curse of dimensionality. We have not found any references in the maintenance literature that answers the question of problem size limits to a satisfactory level.

Quantifying the computational limitations of DP is also important when comparing DP to the alternatives described in Chapter 2, namely policy optimization, heuristics, and approximate MDP methods such as ADP and RL. All of these methods are also hampered by some form of the curse of dimensionality. For instance, once an MDP model has been established, we can turn to ADP to find well-performing policies. Usually, ADP algorithms are tailored to the specific MDP at hand, in order to exploit any structural properties in the MDP. In Jiang and Powell (2015), an ADP algorithm is developed that exploits the monotonicity in the value function to enhance the convergence rate. They test it on a regenerative optimal stopping problem, which can be seen as a finite-horizon CBM problem. The ADP algorithm achieves 90% of the performance of the optimal policy using between one and two orders of magnitude less time compared to DP. However, when the number of components in the problem increases, the time required by ADP and DP grow with the same factor. Approximate algorithms are therefore unfortunately not silver bullets, but in terms of multi-component problems they are able to find good policies in problems that are substantially larger. Regarding policy optimization, we described in Section 2.3.1 how threshold-based policies must be optimized approximately using various metaheuristics when the number of components is large (14 components in the examples in Liu and Huang (2010) and Zhou et al. (2013)). The curse of dimensionality is therefore something that is inherent to the problem, and not to the algorithm for optimization (Sutton and Barto 2018).

In Section 2.2 we gave an overview of the different maintenance modeling aspects. From this we can summarize the system considered in this chapter as a multi-component system with non-repairable components that all deteriorate according to a gamma process. Self-announcing failures occur when component deterioration exceeds a fixed failure limit, L. The system is subject to structural dependence from a K-out-of-N system reliability, stochastic dependence modeled with a Clayton-Lévy copula function, and economic dependence via a joint setup cost for replacement actions. Regarding the available information, we formulate both a TBM MDP model and a CBM MDP model. In the TBM model, we assume that we only know the age of components, and whether or not they are still functioning. In the CBM model we assume that the deterioration level is observed. Hence, the CBM model is the case, where we have full information about the gamma deterioration process of a component, $\{X_t\}_{t\in[0,\infty)}$, and the TBM model then represents a case, where we have full knowledge about the distribution of the failure time $T = \inf\{t \in [0,\infty) : X_t \geq L\}$. We use the Gamma process primarily because this is the most common deterioration process in maintenance optimization papers, and secondly to analyze the combination of a continuous deterioration process and DP optimization.

The main reason we consider both TBM and CBM models, is that the transition probability structure in the two MDPs are quite different, and therefore the computational requirements are also different. Nonetheless, we also see a contribution in the construction of the models themselves. In particular, at the time this chapter was written, the incorporation of continuously deteriorating and stochastically dependent components in an MDP model had not been considered before. In a recent study, Xu et al. 2021, a comprehensive sensitivity analysis is carried out for a CBM model that resembles the one we consider. This includes components deteriorating according to gamma processes with copula dependence. However, the authors do not address the computational aspects, and they also restrict the action space such that an optimal policy is not guaranteed to be identified[1].

The unified view of considering a multi-component system, in which we compute both optimal TBM and optimal CBM policies, is also a novelty. As a side note, this is also the first TBM model for a system with stochastic dependence caused by an external environment (recall Section 2.2.5). This multi-component TBM model may be of limited practical use, since estimating its transition probabilities requires a tremendous amount of data for each possible combination of component failure and the possible combinations of components ages at the moment of failure. However, in Chapter 5 we consider both the CBM and the TBM model again, and compare the performance of the optimal policies obtained from each model. We argue that this comparison is only meaningful if we compare the optimal policies. In Chapter 5, the TBM model therefore has a purpose from a maintenance theoretical perspective.

[1] Components closest to failure must always be replaced before less deteriorated components, and if all components fail they must all be replaced again. In Chapter 4 we describe an example of how it is sometimes optimal to let a component fail and never replace it again.

3.1 Paper A

A numerical study of Markov decision process algorithms for multi-component replacement problems*

Jesper Fink Andersen[a,*], Anders Reenberg Andersen[a], Murat Kulahci[a,b], Bo Friis Nielsen[a]

[a] *Technical University of Denmark, Department of Applied Mathematics and Computer Science, Anker Engelunds Vej 1, 2800 Kgs. Lyngby, Denmark*
[b] *Luleå University of Technology, Department of Business Administration, Technology and Social Sciences, Luleå, Sweden*

Abstract

We present a unified modeling framework for Time-Based Maintenance (TBM) and Condition-Based Maintenance (CBM) for optimization of replacements in multi-component systems. The considered system has a K-out-of-N reliability structure, and components deteriorate according to a multivariate gamma process with Lévy copula dependence. The TBM and CBM models are formulated as Markov Decision Processes (MDPs), and optimal policies are found using dynamic programming. Solving the CBM model requires that the continuous deterioration process is discretized. We therefore investigate the discretization level required for obtaining a near-optimal policy. Our results indicate that a coarser discretization level than previously suggested in the literature is adequate, indicating that dynamic programming is a feasible approach for optimization in multi-component systems. We further demonstrate this through empirical results for the size limit of the MDP models when solved with an optimized implementation of modified policy iteration. The TBM model can generally be solved with more components than the CBM model, since the former has a sparser state transition structure. In the special case of independent component deterioration, transition probabilities can be calculated efficiently at runtime. This reduces the memory requirements substantially. For this case, we also achieved a tenfold speedup when using ten processors in a parallel implementation of algorithm. Altogether, our results show that the computational requirements for systems with independent component deterioration increase at a slower rate than for systems with stochastic dependence.

Keywords: Maintenance, Dynamic programming, Multi-component system, Markov decision process, Numerical study

1. Introduction

In traditional maintenance models, the decision to maintain a system is often based on its age at the time of the decision, also known as Time-Based Maintenance (TBM). Records of failure times of similar systems make it possible to estimate a lifetime distribution that describes the uncertainty in the time to failure within that population of systems. An alternative approach is Condition-Based Maintenance (CBM), where information about the physical condition of the system is utilized for maintenance decisions.

For both TBM and CBM, the methods for optimizing the maintenance policy can be classified using the scheme presented in Nicolai & Dekker (2008). Here, the authors divide optimization procedures into policy optimization, exact-, and heuristic methods. They also classify according to the planning horizon, which can be either infinite or finite. We consider exact methods and an infinite horizon in this study. Policy optimization is the most common approach for infinite planning horizons, where the system dynamics are assumed stationary. A parametrized policy, for instance a variation of a control-limit policy, is considered and its parameters are optimized by deriving an analytical expression for the long-run cost per time unit using renewal theory (Castanier et al., 2005; Abdel-Hameed, 1987; Zhang et al., 2020). The restriction to a specific type of policy facilitates the analyses, but the solution is generally not guaranteed to be globally optimal. An example of a heuristic optimization approach is given in Vu et al. (2014) for a multi-component system, where cost reductions can be obtained by grouping maintenance activities for different components. The approach is to optimize separate infinite-horizon policies for each component and then use a genetic algorithm on a rolling horizon for grouping decisions. Similar to the policy optimization approach, the method proposed by Vu et al. (2014) does not necessarily yield a globally optimal policy, but the heuristic allows for many components to be considered simultaneously. Further examples of policy- and heuristic optimization are found in recent review papers by Olde Keizer et al. (2017) and De Jonge & Scarf (2020).

In both TBM and CBM, a decision is made towards maintaining the system based on new evidence (data) collected from the system. This makes Markov Decision Processes (MDPs), a

⋆Article in press DOI: https://doi.org/10.1016/j.ejor.2021.07.007
⋆Corresponding author
Email addresses: jfan@dtu.dk (Jesper Fink Andersen), arean@dtu.dk (Anders Reenberg Andersen), muku@dtu.dk (Murat Kulahci), bfni@dtu.dk (Bo Friis Nielsen)

modeling framework for sequential decision making, a natural candidate for TBM and CBM models. The methods for optimizing the policy in an MDP can be either Dynamic Programming (DP), which is an exact optimization method, or reinforcement learning and approximate DP, which both belong to the category of heuristic optimization methods. DP algorithms are, under very mild assumptions, guaranteed to find globally optimal solutions in finite time. However, the required computational effort grows exponentially with the dimension of the state space in the MDP. This is known as *the curse of dimensionality in dynamic programming.*

Several MDP models have already been proposed for single-component systems, and because dimensionality is not an issue, these are solved using DP (Elwany et al., 2011; Chen et al., 2015; Neves et al., 2011). In a review on CBM optimization by Alaswad & Xiang (2017) the authors emphasize that as modern industrial systems rarely consist of a single maintainable component, there is an increasing need for multi-component models. In a more recent review on maintenance optimization (De Jonge & Scarf, 2020), it is pointed out that there has been a shift towards heuristic policies and approximate methodologies for multi-component systems. Indeed, reinforcement learning algorithms, even though they are not guaranteed to converge to a globally optimal policy, have been used on both multi-component TBM problems (Xia et al., 2008) and CBM problems (Andriotis & Papakonstantinou, 2019) of massive size.

In this study, we investigate the largest problem size, or correspondingly the maximum number of components, for which it is computationally feasible to obtain a globally optimal solution with DP. We focus on models with replacement as the only maintenance action, fixed regular inspection intervals, and full system observability. In accordance with the motivation for practical use of MDPs given in a recent book (Boucherie & van Dijk, 2017), we give the following two reasons for investigating this:

- Firstly, even though policy optimization and heuristics might scale well for a given multi-component maintenance problem, they should be validated by comparison with optimal solutions to instances of the problem that are ideally as large as possible. This is the only way of strengthening the belief that the heuristics perform well for even bigger problems.

- Secondly, the computational power of a present-day CPU combined with an optimized implementation may be efficient enough for solving real-life multi-component problems of moderate size.

In this study we consider a multi-component system, and formulate two MDP models for the replacement problem for CBM and TBM. The system is general enough that the two resulting models closely resemble most of the MDP models considered in other papers. We solve both MDPs for a varying number of components using DP. We do not discuss the cost performance of the resulting TBM and CBM optimal policies, but consider how the computational requirements for solving the MDPs scale with the number of components. Since the structure of the two MDPs are largely different, the answer is not the same in the two cases.

Single-component CBM studies using MDPs, as proposed in Derman (1963); Kurt & Kharoufeh (2010); Neves et al. (2011), model component deterioration on a discrete set of states $S' = \{0, \ldots, D\}$ with 0 being as good as new and D being a failed component. In TBM modeling, these states represent the age of a component (Dekker et al., 1996; Barreto et al., 2014). When formulating an MDP for a system with N components, a natural choice for the state space, S, would be $S = (S')^N$, which has size $|S| = (D+1)^N$. The numerical examples presented in most multi-component CBM studies are limited to a few components, so that general conclusions from identified structures in the optimal policy and sensitivity analyses are easier to convey. For instance, in Sun et al. (2018) the authors consider a system with $N = 3$ and $D = 19$ as their largest example, which results in $|S| = (19+1)^3 = 8000$. A larger example is found in Jiang & Powell (2015) where a finite-horizon CBM replacement problem with $N = 7$ components, $D = 10$, and a state space size of $|S| = (10+1)^7 \approx 1.9 \times 10^7$ is solved to optimality. Concerning multi-component TBM models using MDP, the largest example we found is in Barreto et al. (2014), where an infinite-horizon TBM replacement problem with up to $N = 5$ components and $|S| \approx 1.6 \times 10^5$ is solved to optimality. The reported computation times in these studies are heavily dependent on the algorithm implementation and the hardware that is used. Their purpose is to act as a baseline for comparison with novel heuristic algorithms, and as such we cannot use them as guidelines for a general size limit of the multi-component maintenance problems. The implementation details are an often overlooked aspect in studies using DP, but they have a crucial impact on the performance. For example, our implementation is able to solve numerical examples from Olde Keizer et al. (2016) and Barreto et al. (2014) orders of magnitude faster, than the times reported in the studies.

Maintenance models using MDP often start by assuming a discrete state space, because this is a requirement of DP algorithms. In practice, the condition monitoring procedure in a CBM

application includes processing raw data signals of physical variables such as vibration or temperature. In this case, the deterioration process of the components is most naturally modeled on a continuous state-space. In accordance, we assume the underlying deterioration of the components follow a gamma process in the system we consider. We then explore the effects of discretization, that is, number of intermediate states between as-good-as-new and failure. The level of discretization greatly affects the number of components we can handle in a multi-component system. Nonetheless, we have not found any studies that go into detail in this aspect. Our results indicate that a coarser discretization level than previously suggested is adequate.

The potential for optimization in an implementation of DP depends on the system characteristics, for instance how we compute and store the transition probabilities in the MDP. In this context, it is important whether the deterioration processes of the system components are dependent, also known as stochastic dependence. In the system we consider, the dependence between the gamma process of each component is modeled using a Lévy Copula. We illustrate how the level of dependence affects the computational requirements for solving the MDP.

In summation, the contributions of our paper are the following:

- We propose a unifying modeling framework for TBM and CBM in multi-component systems.
- We provide realistic limits to the size of multi-component replacement problems, for which they can still be solved to optimality using commonly accessible computer resources.
- We empirically demonstrate the different computational limitations when solving large CBM problems compared to large TBM problems.
- We show that a relatively coarse choice of discretization level is sufficient to solve multi-component CBM problems with continuous-state deterioration processes, such that a near-optimal solution is obtained.
- We show how stochastic dependence among components affects the computational requirements.

The rest of the paper is organized as follows: In Section 2, we present the system and formulate the MDP models for CBM and TBM. In Section 3, we summarize the DP algorithms we use to solve the MDPs, and relevant implementation details. Section 4 contains the results of the numerical experiments, and in Section 5 we provide a conclusion to the study.

2. Problem description

We consider a multi-component system for which we formulate two replacement models for CBM and TBM, both using an MDP. In the models, we assume (a) fixed inspection intervals, (b) full system observability, and (c) perfect maintenance actions, i.e., replacements.

The purpose of this study is to investigate the computational requirements for solving the two models, and we note that relaxing any of the three assumptions makes this task more difficult. Before we proceed to the formulation of the models, we therefore provide some examples of how assumptions (a)–(c) can be relaxed and how this affects the resulting optimization problem.

(a) The inspection frequency, for revealing the system condition, is sometimes assumed to be a decision variable. An example of a policy optimization approach is the delay-time model with delayed postponement in van Oosterom et al. (2014). Here, the policy is parametrized by a threshold for the age at which we inspect a component, and if it is found to be close to failing, a second parameter determines the time until it is replaced. One way to include irregular inspections in an MDP formulation is to consider a very short interval between decision epochs and include "inspection" and "no inspection" as possible actions (Andriotis & Papakonstantinou, 2019). The MDP version then considers a more flexible policy space than the policy optimization version, since an inspection or replacement may be triggered by any combination of the time since the last inspection and the condition found at that inspection. In that sense, this MDP formulation can also be seen as a hybrid of TBM and CBM. However, an obvious caveat is that the state space in this MDP has a higher dimension than a pure CBM or TBM model, since it requires two variables per system component, namely age and condition.

(b) The issue of partial information is due to inspections not being perfect. This can be dealt with by formulating the optimization problem as a partially observable MDP. Except for very small problems, these models can only be solved approximately either using point-based algorithms (Nguyen et al., 2019; Pineau et al., 2003), or a policy optimization approach as in Naderkhani ZG & Makis (2015).

(c) Imperfect maintenance can be incorporated by including actions in the MDP that improve the condition or reduce the age of components, but not all the way to the as-good-as-new state. The result is a larger action space and, if the effect of the imperfect maintenance is random, more nonzero transition probabilities in the MDP. Both of these factors make for a more difficult

optimization problem.

2.1. System description

We consider a multi-component system with N deteriorating components with a K-out-of-N reliability structure, meaning that the system is functioning as long as K components are functioning. We choose this reliability structure as it is quite general and it includes the special cases of series systems ($K = N$) and parallel systems ($K = 1$). We note that other reliability structures, such as series-parallel systems, can be modeled by appropriately changing the reward function in Section 2.3. Examples of other studies that use MDP and the K-out-of-N structure are Sun et al. (2018); Olde Keizer et al. (2016); Andriotis & Papakonstantinou (2019).

The components are subject to deterioration, and their joint condition is described by a multivariate stochastic process $\{\mathbf{X}_t\}_{t\in[0,\infty)} = \{(X_t^1, \ldots, X_t^N)\}_{t\in[0,\infty)}$, with X_t^i being the condition of component i at time t. We assume $X_0^i = 0$ and that component i fails when X_t^i reaches a failure threshold L, which is assumed to be the same for all components without loss of generality for the process we describe in Section 2.2. The components are stochastically dependent, meaning that the marginal processes X_t^i are mutually dependent. This is relevant for systems where the deterioration of components are affected by the same external factors in the operating environment, e.g., weather condition.

In the CBM model, we assume that we observe the process, i.e., the condition of each component. In the TBM model, we only observe whether components are functioning or not, and replacement decisions are based on the ages of the components. We assume that replacements can be carried out at regularly spaced maintenance windows, and that the time required to replace a component is negligible. If component i is replaced before it fails, we incur a preventive replacement cost, c_p^i, and if the replacement happens after the failure we incur a corrective replacement cost, c_c^i. Furthermore, we assume there is a setup cost, c_s, if at least one component is replaced in a given maintenance window and a system failure cost, c_f, if less than K components are functioning at the time of replacement.

2.2. Deterioration process

In practice, CBM often involves monitoring physical variables of the components, hence it is natural to model the deterioration as a continuous-state stochastic process. The choice of stochastic

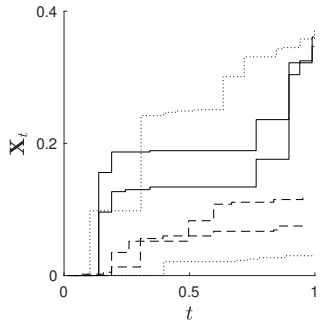


Figure 1: Realizations of $\mathbf{X}_t$ for $N = 2$, $\alpha_1 = \alpha_2 = 7/4$, $\beta_1 = \beta_2 = 15/2$ and $\theta = 0.2$ (dotted), $\theta = 1$ (dashed), $\theta = 3$ (solid).

process is often dictated by the nature of the physical deterioration process, whether it is corrosion, shock damage, crack growth, and the kind of data that is collected, e.g. temperature, vibration magnitude, or geometry. In several CBM studies, MDPs are used in conjunction with such deterioration processes. For example, the inverse Gaussian process (Chen et al., 2015), the gamma process (Nguyen et al., 2019; Andriotis & Papakonstantinou, 2019), Brownian motion with drift (Sun et al., 2018), and geometric Brownian motion (Elwany et al., 2011). Recently, Xu et al. (2021) used the first three of these in a CBM MDP model for a K-out-of-N system like the one considered in this paper. Their focus is on the structure and cost performance of the policies.

In this paper, we assume that $\mathbf{X}_t$ is a multivariate Lévy process, which has time-homogeneous and independent increments $\mathbf{X}_t - \mathbf{X}_s$, $0 \leq s < t$. We assume for each $i = 1, \ldots, N$ that the marginal process is a gamma process, that is, $X_t^i - X_s^i \sim \text{Gamma}((t-s)\alpha_i, \beta_i)$, where α_i and β_i are the shape and rate parameter of component i. The dependence between the components is described via a Clayton-Lévy copula function. This copula has one parameter, $\theta > 0$, which dictates the dependence of jump sizes, where larger values increases the tendency to observe simultaneous large jumps in each component. Figure 1 shows simulated realizations of $\mathbf{X}_t$ with two components for different values of θ. We will also consider the special case of independent components, which we will abbreviate as the $\theta = 0$ case. We will not go into details about Lévy Copulas but refer the reader to Grothe & Hofert (2015), which presents the simulation algorithm we use in Appendix A, or Shi et al. (2020); Li et al. (2016); Jiang et al. (2019), where gamma processes and the Clayton-Lévy copula are used in the context of deterioration modeling.

2.3. MDP formulation

In this section we formulate an MDP for both the CBM and the TBM version of the replacement problem. In both models we consider the maintenance windows, which in MDP terminology are called decision epochs, to be spaced with unit distance, $\tau \in \{0, 1, \ldots\}$. At each epoch, the system occupies one of a finite set of states $S = \{0, \ldots, D\}^N$, where an element $\mathbf{s} = (s_1, \ldots, s_N)$ describes the state of the system. In both the CBM and the TBM model, the state of component i is a number $s_i \in \{0, \ldots, D\}$, where $s_i = 0$ corresponds to a new component and $s_i = D$ is a failed component. The information in s_i is different for the two models though. In the CBM model, s_i is a discretized version of the condition, X_τ^i, and in the TBM model s_i is the age of the component, i.e., the number of epochs since the last replacement.

At each decision epoch we choose an action from a finite set $A = \{0, 1\}^N$. When an element a_i of an action $\mathbf{a} \in A$ equals 1 (0), this corresponds to replacing (not replacing) component i. Depending on the current state $\mathbf{s} \in S$ and action $\mathbf{a} \in A$ we receive a reward $r(\mathbf{s}, \mathbf{a})$ and the system transitions to a new state $\mathbf{s}' \in S$ with probability $p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$. Together S, A , $p(\cdot|\cdot, \cdot)$, and $r(\cdot, \cdot)$ define an MDP. Actions are chosen according to a policy, $\pi \in \Pi$, where Π is the set of all mappings from S to A. Solving the MDP means to identify an optimal policy, $\pi^* \in \Pi$, that maximizes the reward received over an infinite horizon. The optimal policy π^* minimizes the long-run maintenance cost, which we formulate as a maximization problem where all rewards in the MDP are negative.

The reward function is the same for both the TBM and the CBM model and it is defined by

$$r(\mathbf{s}, \mathbf{a}) = \sum_{i=1}^{N} a_i \left(c_p^i \mathbf{1}_{s_i<D} + c_c^i \mathbf{1}_{s_i=D} \right) + c_s \left(1 - \prod_{i=1}^{N} (1 - a_i) \right) + c_f \mathbf{1}_{|\{i:s_i<D\}|<K}, \tag{1}$$

where $\mathbf{1}_A$ is the indicator function for event A. The first term of Equation (1), is the replacement costs. The second and third terms account for the setup cost and system failure cost, respectively.

In Sections 2.4 and 2.5 we describe, for each of the two models, how we obtain transition probabilities $p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$ from the underlying deterioration process $\mathbf{X}_t$. Since the replacement of a component is assumed to be instantaneous, the replacement action $\mathbf{a}$ instantly moves the system from state $\mathbf{s}$ to a *post-decision* state, $((1 - a_1)s_1, \ldots, (1 - a_N)s_N) \in S$, that is, the same state only with zero entries for the replaced components. The transition to state $\mathbf{s}'$ then occurs with probability $p(\mathbf{s}'|((1 - a_1)s_1, \ldots, (1 - a_N)s_N), \mathbf{0})$, where $\mathbf{0} \in A$ is the action of not replacing any components. Let $q(\mathbf{s}'|\mathbf{s}) = p(\mathbf{s}'|\mathbf{s}, \mathbf{0})$. It now suffices to specify $q(\mathbf{s}'|\mathbf{s})$ for all $\mathbf{s}, \mathbf{s}' \in S$, which is determined from the evolution of $\mathbf{X}_t$.

The process $\mathbf{X}_t$ describes the condition of the unmaintained system, where no replacements are performed. The condition of the maintained system, here denoted $\mathbf{Y}_t = (Y_t^1, \ldots, Y_t^N)$ is the process, with the same increments $\mathbf{Y}_{t_2} - \mathbf{Y}_{t_1} = \mathbf{X}_{t_2} - \mathbf{X}_{t_1}$, $0 \leq t_1 < t_2 < \infty$, when no replacements are done in the interval $[t_1, t_2]$, and for which a replacement, say of component i at time t, leads to $Y_t^i = 0$. The time of replacements depends on the policy, π being used. Specifically, at epoch τ, the true system condition is mapped to an element of the discrete MDP state space, $\mathbf{Z}_\tau = (Z_\tau^1, \ldots, Z_\tau^N)$, and $\mathbf{Z}_\tau$ is then mapped to a replacement action via π. The mapping to $\mathbf{Z}_\tau$ is defined differently for the CBM model and the TBM model. In Section 2.6, we discuss the implications of using $\mathbf{Z}_\tau$ to make decisions.

2.4. CBM model

For the CBM model, a state $\mathbf{s} \in S$ is the condition information from the process $\mathbf{Y}_t$, only in a discretized form. We could formulate the MDP using the same continuous state space as the process $\mathbf{Y}_t$. Such a model can be useful if the objective is to prove that the optimal policy has a specific structure, which is done in Elwany et al. (2011); Chen et al. (2015); Sun et al. (2018); Özekici (1988). Identifying the optimal policy in an MDP with a continuous state-space is equivalent to solving a nonlinear functional equation, namely the Bellman equation. But, even with a characterization of the optimal policy, it is generally impossible to solve this equation analytically (Özekici, 1988). The MDP must therefore be solved using the iterative algorithms in Section 3, but in order to do so, the state space must first be discretized and transition probabilities between the new discrete states must be approximated. Besides the references already mentioned in this paragraph, Andriotis & Papakonstantinou (2019); Nguyen et al. (2019); Olde Keizer et al. (2016) also discretize a continuous deterioration process in order solve an MDP.

Recall that L denotes the failure limit for each of the marginal deterioration processes. We discretize the interval $[0, L)$ into D equally sized intervals $I_k = [kL/D, (k+1)L/D)$, $k = 0, \ldots, D-1$, and let $I_D = [L, \infty)$. We then form a mapping of the true condition of component i at epoch τ, Y_τ^i, to a discrete state $s_i \in \{0, \ldots, D\}$, by letting $Z_\tau^i = s_i$, if $Y_\tau^i \in I_{s_i}$. Thus, $\mathbf{Y}_\tau \in \mathbf{I_s}$ where $\mathbf{I_s} = I_{s_1} \times \ldots \times I_{s_N}$ is mapped to $\mathbf{Z}_\tau = \mathbf{s}$. In the MDP, we approximate the transition probabilities, $q(\mathbf{s}'|\mathbf{s})$, between the discrete states $\mathbf{s}, \mathbf{s}' \in S$ by

$$q(\mathbf{s}'|\mathbf{s}) = P(\mathbf{X}_{\tau+1} \in \mathbf{I}_{\mathbf{s}'} | X_\tau^i = s_i L/D, i = 1, \ldots, N), \tag{2}$$

i.e., we assume component i is at the left endpoint of interval I_{s_i} at epoch τ.

In the case of independent component deterioration ($\theta = 0$), the probability in Equation (2) factorizes. Let $q_i(s'|s) = P(X^i_{\tau+1} \in I_{s'} | X^i_\tau = sL/D)$, $s, s' \in \{0, \ldots, D\}$, i.e. the probability that component i moves from discrete state s to s'. Then

$$q(\mathbf{s}'|\mathbf{s}) = \prod_{i=1}^{N} q_i(s'_i|s_i). \tag{3}$$

We can calculate $q_i(s'_i|s_i)$ from the distribution function of the marginal process X^i_t. When $\theta > 0$ there is, however, no known analytical expression for the distribution of $\mathbf{X}_t$, so we resort to Monte-Carlo simulation to estimate the probabilities in Equation (2). The details of this procedure are described in Appendix A.1. We note that for other deterioration processes with stochastic dependence, the MDP transition probabilities can be calculated without the need for Monte Carlo methods. In Shi et al. (2020) an analytical expression for the joint distribution function is given for an α-stable deterioration process with Lévy copula dependence. In Zhang et al. (2020) stochastic dependence is modeled by having a failure of one component induce shock damage to other components.

2.5. TBM model

A number of studies consider multi-component TBM models similar to the MDP we formulate in this section, namely Haurie & l'Ecuyer (1982); Dekker et al. (1996); Sun et al. (2007); Xia et al. (2008); Barreto et al. (2014). In a TBM model, the element s_i of a state $\mathbf{s} \in S$ represents the age of component i. At each transition, a component can either fail and transition to state D, or age by one time unit, thereby transitioning to state $s_i + 1$. In the mentioned references, the failure probability is assumed to be a known function of the component age. In our model, we construct the failure probabilities from the underlying unobserved deterioration process. That is, $Z^i_\tau = s_i$ if $Y^i_\tau < L$ and the last replacement was at epoch $\tau - s_i$. If the component has failed, $Y^i_\tau \geq L$, or the last replacement was at least D epochs ago, then $Z^i_\tau = D$. Therefore, D is also a truncation point for the maximum age of a component, because the algorithms we describe in Section 3 only work for finite state spaces. We want to set D at a level high enough that the components are very unlikely to reach that age. On the other hand, we do not want the state space to be larger than necessary, so D is set individually for each component, D_i, $i \in \{1, \ldots, N\}$, such that $P(X^i_{D_i} < L)$ is close to zero. The state space of the TBM MDP is therefore $S = \{0, \ldots, D_1\} \times \ldots \times \{0, \ldots, D_N\}$.

We first look at the transition probabilities of a system with independent component deterioration ($\theta = 0$) as these are easier to express. Let $q_i(s'|s)$ denote the transition probability of a single component as in Section 2.4. For the TBM model we define it as

$$q_i(s'|s) = \begin{cases} P(X^i_{s+1} \geq L | X^i_s < L) & s' = D_i, s < D_i - 1 \\ P(X^i_{s+1} < L | X^i_s < L) & s' = s + 1 < D_i \\ 1 & D_i - 1 \leq s \leq s' = D_i \\ 0 & \text{else.} \end{cases} \tag{4}$$

The first line is the probability of a functioning component of age s failing, the second line is the probability of not failing. If component i has age $s = D_i - 1$ or is failed, $s = D_i$, it will be in state D_i at the next epoch with certainty, which is the content of the third line. For $\theta = 0$, the joint transition probability $q(\mathbf{s}'|\mathbf{s})$ can again be calculated using Equation (3).

Now consider $\theta > 0$ and that the system is in (post-decision) state $\mathbf{s} \in S$. We define $F_\mathbf{s}, E_\mathbf{s} \subseteq \{0, \ldots, N\}$ by $F_\mathbf{s} = \{i : s_i = D_i\}$ and $E_\mathbf{s} = \{i : s_i = D_i - 1\}$, i.e., the set of failed and almost failed components, respectively. The only possible transitions are to the states $\mathbf{s}' \in S$ for which $F_\mathbf{s} \cup E_\mathbf{s} \subseteq F_{\mathbf{s}'}$ and $s'_i = s_i + 1$ for $i \in F^\mathsf{c}_{\mathbf{s}'}$, where A^c denotes the set complement of A. In other words, each working component in state $\mathbf{s}$ has the possibility of failing, and there is one state, $\mathbf{s}'$, for each combination of possible failures. We approximate this transition probability with

$$q(\mathbf{s}'|\mathbf{s}) = P\left(\left(\bigcap_{i \in F_{\mathbf{s}'} \setminus (F_\mathbf{s} \cup E_\mathbf{s})} Y^i_{\tau+1} \geq L \right) \cap \left(\bigcap_{i \in F^\mathsf{c}_{\mathbf{s}'}} Y^i_{\tau+1} < L \right) \middle| \bigcap_{i \in (F_\mathbf{s} \cup E_\mathbf{s})^\mathsf{c}} Y^i_\tau < L \right), \tag{5}$$

where the policy that decides replacement times in $\mathbf{Y}_t$ are described together with the Monte Carlo estimation procedure in Appendix A.2. The set $F_{\mathbf{s}'} \setminus (F_\mathbf{s} \cup E_\mathbf{s})$ in Equation (5) are the functioning components that fail by exceeding the limit L in the transition to $\mathbf{s}'$. The appearance of this set is because we do not require the components in $F_\mathbf{s} \cup E_\mathbf{s}$ to exceed the failure limit L as they are by definition already certain to be in the failed state after the transition. The set $F^\mathsf{c}_{\mathbf{s}'}$ are the components that are still functioning in state $\mathbf{s}'$. In Section 2.6 we elaborate on why Equation (5) is only an approximation and the implications for the policy we obtain by solving the MDP.

2.6. Markov properties of induced stochastic processes

Globally optimal policies can be obtained via MDPs and DP, as we mentioned in the introduction. However, due to the assumed deterioration process, we can only obtain close approximations

to a globally optimal policy for the system we consider. For the TBM case this is a consequence of the Lévy copula dependence between components, and in the CBM case it is because of the discretization of the state space. We provide further details in this section.

In an MDP, any policy, $\pi : S \to A$, induces a Markov chain, $\mathbf{S}_\tau$, $\tau = 0, 1 \ldots$, on the set of states S, where $\mathbf{S}_\tau$ is the state at epoch τ. Specifically, $\mathbf{S}_\tau$ has the Markov property since $P(\mathbf{S}_{\tau+1} = \mathbf{s}'|\mathbf{S}_\tau = \mathbf{s}) = p(\mathbf{s}'|\mathbf{s}, \pi(\mathbf{s}))$, which does not depend on $\mathbf{S}_{\tau-1}, \ldots, \mathbf{S}_0$. Recall that $\mathbf{Z}_\tau$ is the stochastic process obtained by mapping the true system condition, $\mathbf{Y}_\tau$, to the MDP state space S. The transition probabilities, $p(\mathbf{s}'|\mathbf{s}, \pi(\mathbf{s}))$ defined in Equations (2) and (5), are approximations of the probabilities $P(\mathbf{Z}_{\tau+1} = \mathbf{s}'|\mathbf{Z}_\tau = \mathbf{s})$, $\mathbf{s}, \mathbf{s}' \in S$; thus, the processes $\mathbf{Z}_\tau$ and $\mathbf{S}_\tau$ have different properties. This is demonstrated in Sections 2.6.1 and 2.6.2 through examples showing that $\mathbf{Z}_\tau$ is not necessarily Markovian.

2.6.1. TBM

Consider the case of a two-component system where the policy is to never replace any components. Suppose we have observed the following: At the beginning of the previous epoch, both components were new, $\mathbf{Z}_{\tau-1} = (0, 0)$, and at the current epoch the first component has failed while the second has aged by one, $\mathbf{Z}_\tau = (D_1, 1)$. The failure is caused by a large increment, $Y^1_\tau - Y^1_{\tau-1}$, in the underlying deterioration process. Since this is correlated with $Y^2_\tau - Y^2_{\tau-1}$, knowing $\mathbf{Z}_{\tau-1}$ provides additional information about how likely the second component is to fail before the next epoch, $\tau + 1$, so it is possible that $P(\mathbf{Z}_{\tau+1} = (D_1, D_2)|\mathbf{Z}_\tau = (D_1, 1))$ and $P(\mathbf{Z}_{\tau+1} = (D_1, D_2)|\mathbf{Z}_\tau = (D_1, 1), \mathbf{Z}_{\tau-1} = (0, 0))$ are not equal.

The implication is that in a multi-component system with dependent deterioration increments among components, the globally optimal TBM policy is a history-dependent policy (Puterman, 2005). This means that actions are chosen based on the entire history of observed states, and finding the optimal policy within this class is generally computationally intractable. The MDP we propose in Section 2.5 is therefore only an approximate model. Nonetheless, we include it in the numerical study for two reasons: First, the policy obtained from solving the MDP is at least as good as any heuristic policy that is contained in the set of Markovian policies, e.g., age-based and block-replacement policies. Secondly, from a computational aspect the case with dependent components is interesting as it represents the situation that all transition probabilities in the MDP have to be estimated in advance, which requires additional memory. Finally, it is important to

notice that when components deteriorate independently, the issues regarding the Markov property disappear, since the MDP transition probabilities in Equation (4) are not approximations as in Equation (5) but exact.

2.6.2. CBM

In the CBM case, it is possible to construct a two-component example similar to that in Section 2.6.1, to show that the process $\mathbf{Z}_\tau$ is non-Markovian. However, this is also true even in a single-component system. Suppose we observe $\mathbf{Z}_{\tau-2} = 0$, $\mathbf{Z}_{\tau-1} = 0$, and $\mathbf{Z}_\tau = 0$, which from the definitions of $\mathbf{Z}_\tau$ and S imply that $Y_0^1, Y_1^1, Y_2^1 \in [0, L/D)$. Knowing that Y_τ^1 spent two periods in $[0, L/D)$ provides additional information about whether it will stay in this interval, so in general $P(\mathbf{Z}_{\tau+1} = 0|\mathbf{Z}_\tau = 0)$ and $P(\mathbf{Z}_{\tau+1} = 0|\mathbf{Z}_\tau = \mathbf{Z}_{\tau-1} = \mathbf{Z}_{\tau-2} = 0)$ are not equal. As the example illustrates, the non-Markovian behavior of the observed MDP states, $\mathbf{Z}_\tau$, stems from the discretization of the continuous state space. As we increase the number intervals, D, the error introduced by the discretization diminishes, hence the policy we obtain from the MDP will approach a globally optimal CBM policy.

3. Solution methods

Our overall goal is to solve as large instances of the MDPs formulated in Section 2 as possible. To solve the MDPs, we use iterative DP algorithms. There are two different avenues of optimization for these algorithms: Lowering the required number of iterations, and optimizing the speed of the calculations within each iteration. The latter mainly revolves around how transition probabilities are handled in the implementation. Details regarding this are provided in Section 3.7.

The former can be achieved by choosing the best algorithm configuration from a toolbox of methods. These are presented in Sections 3.1–3.6. The three main algorithms, Value Iteration (VI), Policy Iteration (PI), and Modified Policy Iteration (MPI) are briefly reviewed in Sections 3.1–3.3. It is generally impossible to know in advance, which of the three algorithms is better for a particular MDP. Complexity results for MPD algorithms are surveyed in Littman et al. (1995), however, the authors conclude with the statement that these results are of marginal use to practitioners. In practice, the algorithms are usually much faster than their theoretical worst-case run times (Sutton & Barto, 2018). The empirical approach we employ is therefore justified. Besides iterative algorithms, optimal policies can also be obtained via linear programming, but

this method becomes impractical already at much smaller problem sizes (Sutton & Barto, 2018), and is therefore not included.

We solve the infinite-horizon version of the problems using the expected total discounted reward optimality criterion. That is, from the set of all mappings from S to A, Π, we seek to find the optimal policy, $\pi^* \in \Pi$, satisfying $v_{\pi^*}(\mathbf{s}) = \max_{\pi \in \Pi} v_\pi(\mathbf{s})\ \forall \mathbf{s} \in S$, where $v_\pi : S \to \mathbb{R}$ is called the value function and where $v_\pi(\mathbf{s})$ is the expected total discounted reward when starting in state $\mathbf{s} \in S$ and following policy $\pi \in \Pi$. The optimal policy is found by solving the Bellman equations,

$$v(\mathbf{s}) = \max_{\mathbf{a} \in A} \left\{ r(\mathbf{s}, \mathbf{a}) + \gamma \sum_{\mathbf{s}' \in S} p(\mathbf{s}'|\mathbf{s}, \mathbf{a}) v(\mathbf{s}') \right\} \quad \forall \mathbf{s} \in S, \tag{6}$$

where $0 \leq \gamma < 1$ is the discount factor.

3.1. Value iteration

The standard value iteration algorithm is an iterative algorithm that in each iteration, n, updates an estimate of the value function, $v_n \in \mathbb{R}^{|S|}$. Starting with an arbitrary v_0 the updates are computed by

$$v_{n+1}(\mathbf{s}) = \max_{\mathbf{a} \in A} \left\{ r(\mathbf{s}, \mathbf{a}) + \gamma \sum_{\mathbf{s}' \in S} p(\mathbf{s}'|\mathbf{s}, \mathbf{a}) v_n(\mathbf{s}') \right\} \quad \forall \mathbf{s} \in S. \tag{7}$$

When Equation (7) is used repeatedly $v_n(\mathbf{s}) \to v_{\pi^*}(\mathbf{s})$ as $n \to \infty$, see Theorem 6.3.1 in Puterman (2005). When the value function converges, the final policy is obtained from the maximizing actions in the last use of Equation (7). We elaborate more on convergence in Section 3.5.

3.2. Policy iteration

PI is based on two fundamental steps: *Policy evaluation* in which the value function of the current best policy is computed, and *Policy improvement*, in which the policy is improved based on the recently computed value function.

Let π_n denote the policy in iteration n, and let v_n denote the value function that corresponds to policy π_n. The purpose of the first step of PI is to derive v_n by solving the linear system $(I - \gamma P_{\pi_n}) v_n = r_{\pi_n}$. Here, $r_{\pi_n} \in \mathbb{R}^{|S|}$ is the reward vector with elements $r(\mathbf{s}, \pi_n(\mathbf{s}))$ and $P_{\pi_n} \in \mathbb{R}^{|S| \times |S|}$ is the transition probability matrix with elements $p(\mathbf{s}'|\mathbf{s}, \pi_n(\mathbf{s}))$, $\mathbf{s}, \mathbf{s}' \in S$. Subsequently, the second step of PI finds an improved policy by applying Equation (7) and assigning the maximizing actions to π_{n+1}. The algorithm is initiated with an arbitrary $\pi_0 \in \Pi$ and continues until an

improving policy can no longer be obtained — i.e., when $\pi_{n+1} = \pi_n$. Because the problem instances we wish to solve are quite large, we solve the linear system $(I - \gamma P_{\pi_n})v_n = r_{\pi_n}$ using iterative methods. Letting $\hat{v}_0$ be the value function after using Equation (7) in the improvement step we then produce iterates $\hat{v}_k$ that approach v_n as $k \to \infty$ in the following manner,

$$\hat{v}_{k+1}(\mathbf{s}) = r(\mathbf{s}, \pi_n(\mathbf{s})) + \gamma \sum_{\mathbf{s}' \in S} p(\mathbf{s}'|\mathbf{s}, \pi_n(\mathbf{s}))\hat{v}_k(\mathbf{s}') \quad \forall \mathbf{s} \in S. \tag{8}$$

In practice, the required number of policy improvement steps can be quite low, and if this is the case, PI is often superior to VI. However, for some MDPs it may be inefficient to perform an exact policy evaluation at each iteration, in particular if the number of actions is low, because the VI update in Equation (7) is then not much more expensive than the policy evaluation step in Equation (8). The next algorithm can be seen as a combination of VI and PI and attempts to incorporate the advantages of both algorithms.

3.3. Modified policy iteration

We do not require an exact estimate of the value function v_n to find a better policy in the improvement step of PI. If we terminate the policy evaluation procedure in Equation (8) prematurely, we get the MPI algorithm (Puterman, 2005). The evaluation step is now referred to as partial evaluation, and in our implementation we stop at $k = m$ or when the sequence $\hat{v}_k$ converges to v_n, whichever comes first. The optimal iteration limit, $m \in \mathbb{N}$, is determined experimentally. The MPI algorithm terminates when the value function converges upon being updated to $\hat{v}_0$ in the improvement step.

3.4. Update schemes

The updates of the value function in Equations (7) and (8) will be referred to as standard (STD) in Section 4. Besides this, we test two additional update schemes, namely the Gauss-Seidel (GS) method and Successive Over-Relaxation (SOR). Both schemes speed up the convergence of the algorithms by inserting updated values $\hat{v}_{k+1}(\mathbf{s})$ ($v_{n+1}(\mathbf{s})$) into Equation (8) (Equation (7)) as soon as they become available. The SOR method involves a relaxation parameter, $1 \leq \omega < 2$, and the best value must be determined experimentally.

3.5. Stopping criteria

All three algorithms that were presented in Sections 3.1–3.3 require a definition of convergence to be able to stop. In this study, we test two different methods: The supremum norm, $\|v_{n+1}-v_n\| = \max_{\mathbf{s}\in S}|v_{n+1}(\mathbf{s}) - v_n(\mathbf{s})|$, and the span seminorm, $sp(v_{n+1} - v_n) = \max_{\mathbf{s}\in S}\{v_{n+1}(\mathbf{s}) - v_n(\mathbf{s})\} - \min_{\mathbf{s}\in S}\{v_{n+1}(\mathbf{s}) - v_n(\mathbf{s})\}$. Let $\epsilon > 0$ denote a tolerance parameter. When using the supremum norm, we stop the algorithm when $\|v_{n+1} - v_n\| < \epsilon(1-\gamma)/2\gamma$ and for the span seminorm we stop when $sp(v_{n+1} - v_n) < \epsilon(1-\gamma)/\gamma$. This ensures that $\|v_{\pi_{n+1}} - v_{\pi^*}\| < \epsilon$, see Theorem 6.3.1 and Proposition 6.6.5 in Puterman (2005). The span criterion is more sensitive and often terminates the algorithm much earlier. However, it does not apply to the GS and SOR update, which is why we consider both criteria.

3.6. Initializations

The initialization v_0, and π_0 in PI and MPI, affects the number of iterations that are required for algorithms to terminate. In order to assess the effect of the initialization, we consider three different strategies. First, an initialization above the optimal value function, $v_{\pi^*} \leq v_0$, which is simply a zero initialization $v_0(\mathbf{s}) = 0$ and $\pi_0(\mathbf{s}) = \mathbf{0}$ for all $\mathbf{s} \in S$. Secondly, an initialization below the optimal value function, $v_{\pi^*} \geq v_0$, which is formed by setting $\pi_0(\mathbf{s}) = \text{argmax}_{\mathbf{a}}\{r(\mathbf{s},\mathbf{a})\}$ and $v_0(\mathbf{s}) = r(\mathbf{s},\pi_0(\mathbf{s})) + \gamma(1-\gamma)^{-1}\min_{\mathbf{s}}\{r(\mathbf{s},\pi_0(\mathbf{s}))\}$ for all $\mathbf{s} \in S$. Thirdly, we also use a random initialization where $\pi_0(\mathbf{s})$ is sampled randomly from A and $v_0(\mathbf{s})$ is sampled randomly between the upper and lower bound from the other two initializations.

3.6.1. Multigrid algorithm

In the CBM model we also consider a multigrid algorithm similar to that in Chow & Tsitsiklis (1991), where the MDP is solved multiple times for successively finer discretization levels. We first solve the MDP with a small value of D (coarse grid), and use the resulting value function, v^D, and policy, π^D, to initialize the algorithm for solving the MDP with a $2D$ discretization (finer grid), that is, where the length of each interval $I_k \subset [0, L)$ defined in Section 2.4 has been halved. More formally, just as we defined S and the regions $\mathbf{I}_\mathbf{s} \subset [0, L)^N$ for $\mathbf{s} \in S$ for the discretization level D, we let S' and $\mathbf{I}'_{\mathbf{s}'}$ for $\mathbf{s}' \in S'$, be defined correspondingly for the discretization level 2D. Then for each $\mathbf{s}' \in S'$ there is exactly one state $\mathbf{s} \in S$ such that $\mathbf{I}'_{\mathbf{s}'} \subset \mathbf{I}_\mathbf{s}$, and we initialize the MDP for 2D discretization with $v_0(\mathbf{s}') = v^D(\mathbf{s})$ and $\pi_0(\mathbf{s}') = \pi^D(\mathbf{s})$. After solving this MDP, the whole process is repeated until a policy for a sufficiently fine discretization has been obtained.

3.7. Implementation details

In this section we describe different strategies for handling transition probabilities in the algorithm implementation, as this is the main performance bottleneck when solving large MDPs.

3.7.1. Store in memory

The most efficient method in terms of speed is to calculate all transition probabilities $q(\mathbf{s}'|\mathbf{s})$, for all $\mathbf{s}', \mathbf{s} \in S$ where $q(\mathbf{s}'|\mathbf{s}) > 0$, and store them in memory before we run the algorithm. For the TBM model the memory requirement for this is in the order of $|S|2^N$ because it requires storing a probability for each combination of failures of the N components for each state $\mathbf{s} \in S$. For the CBM model the number of nonzero transition probabilities is in the order of $|S|^2$. However, since the deterioration process $\mathbf{X}_t$ is assumed to have stationary increments, we can choose to only store $|S|$ probabilities at the expense of additional computations. For $\mathbf{u} \in S$, $q(\mathbf{u}|\mathbf{0}) = P(\mathbf{X}_1 \in \mathbf{I}_\mathbf{u})$ is the probability that each component advances u_i discrete states in one transition. We define the set

$$U_{\mathbf{s},\mathbf{s}'} = \{\mathbf{u} \in S : u_i + s_i \geq D \text{ if } s_i \leq s'_i = D \text{ and } u_i + s_i = s'_i \text{ if } s_i \leq s'_i < D\}, \tag{9}$$

which are the possible increments that moves us from state $\mathbf{s}$ to $\mathbf{s}'$. Note that if $s'_i = D$, then any increment $u_i \in \{0, \ldots, D\}$ such that $s_i + u_i \geq D$ will result in $s'_i = D$. The probability $q(\mathbf{s}'|\mathbf{s})$ can now be calculated at runtime as the sum

$$q(\mathbf{s}'|\mathbf{s}) = \sum_{\mathbf{u} \in U_{\mathbf{s},\mathbf{s}'}} q(\mathbf{u}|\mathbf{0}), \tag{10}$$

provided we calculate and store $q(\mathbf{s}|\mathbf{0})$ for all $\mathbf{s} \in S$ before running the algorithm. The simulation procedure for estimating $q(\mathbf{s}|\mathbf{0})$ is provided in Appendix A.1.

3.7.2. Calculate at runtime

At some point, when the number of components is large enough, storing all probabilities is impossible. Whether or not we can still solve the MDP depends on how fast we can compute the transition probabilities as they appear in the algorithm at runtime. For the case of dependent components, we would have to do this with Monte-Carlo estimation, which is extremely slow. In the case of independent components it can be done quite efficiently using Equation (3), and it only requires storing $q_i(s'_i|s_i)$ for each $s', s \in \{0, \ldots, D\}$ and $i \in \{1, \ldots, N\}$ which is $(D+1)^2 N$ probabilities.

When updating the value function at a system state $\mathbf{s} \in S$, as in e.g. Equation (7), we need the transition probabilities for all states $\mathbf{s}, \mathbf{s}' \in S$ for which $q(\mathbf{s}'|\mathbf{s}) > 0$. Suppose $q(\mathbf{s}'|\mathbf{s}) > 0$, $q(\mathbf{s}''|\mathbf{s}) > 0$, and that $s'_j \neq s''_j$ for some j but $s'_i = s''_i$ for $i \neq j$. If we have already calculated $q(\mathbf{s}'|\mathbf{s})$ then we have $q(\mathbf{s}''|\mathbf{s}) = q(\mathbf{s}'|\mathbf{s})q_j(s''_j|s_j)/q_j(s'_j|s_j)$, which is faster than calculating Equation (3) from scratch. In our implementation, we use a lexicographical ordering of the states in S to systematically go through all possible transitions knowing which components are identical in $\mathbf{s}'$ and $\mathbf{s}''$, and thereby avoid a substantial number of unnecessary operations.

4. Numerical study

The difficulty of solving the TBM and CBM models described in Section 2 might depend on the parameters of the system. For each number of components, N, we therefore consider three levels of component dependence: Independence ($\theta = 0$), weak dependence ($\theta = 0.2$), and strong dependence ($\theta = 3.0$). For each combination of θ and N we then generate 30 system instances with varying deterioration and cost parameters chosen as follows: K is uniformly distributed on $\{1, \ldots, N\}$, $L = 1$, $\alpha_i \sim \mathcal{U}(13/8, 15/8)$, $\beta_i \sim \mathcal{U}(25/4, 35/4)$, $c_s = -25 - 5(N - K)$, $c_f = -500 - 500(N - K)$, $c_p^i \sim \mathcal{N}(-6\beta_i/\alpha_i, 5^2)$, $c_c^i \sim \mathcal{N}(-12\beta_i/\alpha_i, 5^2)$, where $\mathcal{U}(a, b)$, and $\mathcal{N}(\mu, \sigma^2)$ denote the uniform distribution on the interval $[a, b]$ and the Normal distribution with mean μ and variance σ^2, respectively. We found these parameter ranges, by experimenting with different configurations for $N = 2$ and $N = 3$, and inspecting the resulting optimal policy. If the parameters are not balanced, the resulting optimal policy might be trivial, e.g. $\pi^*(\mathbf{s}) = \mathbf{0}$ for all $\mathbf{s} \in S$, which can be identified very quickly, and we want to avoid these uninteresting cases.

For each value of N, we now have 90 MDPs for CBM and 90 MDPs for TBM. The transition probabilities in the CBM models are estimated from 10^9 realizations of $\mathbf{X}_1$, and in the TBM models we estimate from 10^8 trajectories of $\mathbf{X}_t$ as described in Appendix A. In the TBM models, the age truncation, D_i, for component i, is set as the lowest integer such that $P(X^i_{D_i} < L) < 10^{-6}$. For the chosen distributions of α_i and β_i, this results in D_i being in the range of 12 to 17.

All numerical experiments were performed on a Huawei XH620 V3 server node, which has two Intel Xeon Processor 2660v3 with ten 2.60GHz cores each (we only utilize one core unless explicitly mentioned) and 128GB RAM. All the algorithms were implemented in C++.

4.1. Algorithm comparison

In this section we test each configuration of the solution methods presented in Sections 3.1–3.6. The objective is to identify a configuration that is consistently fast across all system instances.

We test all feasible configurations of algorithm (VI, PI, MPI), value function update (STD, GS, SOR), stopping criterion (supremum, span), MPI iteration limit $m \in \{10, 20, \ldots, 100\}$, SOR relaxation $\omega \in \{1.0, 1.1, \ldots, 1.9\}$, and initialization (above, below, and one random per MDP). The number of feasible combinations is 468 for the TBM model, and 624 for the CBM model because the CBM model also includes the multigrid initialization method. The discount factor and the tolerance parameter are set to $\gamma = 0.99$ and $\epsilon = 0.001$, respectively.

Testing all configurations is too time consuming for the MDPs where $|S|$ is large. Initially we therefore focus on a set of smaller MDPs, where the fastest algorithm configuration for each MDP uses between 10 seconds and 250 seconds to solve it. In the TBM case, we have 51 MDPs with $N = 4$ and $|S|$ between 57,344 and 143,640. In the CBM case, we have 318 MDPs with combinations of $N \in \{2, 3, 4, 5\}$ and $D \in \{4, 6, 8, 12, 16, 24\}$ resulting in $|S|$ between 2197 and 15625.

Table 1 shows the best performing configurations, and demonstrates that the fastest configuration is not the same for all MDPs. In the TBM case, no configuration is the fastest in more than 7 out of the 51 MDPs. Furthermore, any one configuration is on average at least 30% slower when comparing its runtime to the lowest runtime among all configurations. We also note that among the fastest configurations for each MDP, the initialization methods "below", "above", and "random" were present 68%, 28%, and 4% of the time, respectively. Furthermore, for the MPI-m-STD-span-(...) configurations the average increase in runtime from the worst initialization to the best initialization increases with m, from 9% at $m = 20$ to 21% at $m = 90$. Taking all of this into account, we choose the **MPI-60-STD-span-below** configuration when solving larger system instances in the Section 4.3.

In the CBM case, we pick the **MPI-20-STD-span-multigrid** configuration for solving large MDPs. For all combinations of MDP and algorithm configuration, the multigrid initialization is the fastest 74% of the time. In these 74% of the combinations, the second fastest initialization is 60% slower on average. In comparison, when the multigrid initialization is not the fastest, it is 17% slower on average. In 7 out of the 318 CBM MDPs a configuration using the SOR update scheme

was the fastest. In these 7 cases, the MPI-20-STD-span-multigrid configuration is 78% slower on average, so for a given set of system parameters there is a small chance that an SOR configuration will be most effective. However, in many of the MDPs the SOR configurations do not converge, and when they do, they are between 8 and 16 times slower than the fastest configuration.

TBM: 51 MDPs	Configuration	Relative runtime	#fastest
Top 3 relative runtime	MPI-80-STD-span-above	+30.0%	0
	MPI-80-STD-span-below	+30.1%	1
	MPI-90-STD-span-below	+30.2%	3
Top 3 #fastest	MPI-40-STD-span-below	+39.0%	7
	MPI-100-SOR-1.0-sup-below	+54.0%	7
	MPI-60-STD-span-below	**+32.0%**	**4**
CBM: 318 MDPs			
Top 3 relative runtime	MPI-30-STD-span-multigrid	+16.0%	35
	MPI-20-STD-span-multigrid	**+16.3%**	**42**
	MPI-40-STD-span-multigrid	+17.3%	15
Top 3 #fastest	MPI-10-STD-span-multigrid	+19.0%	186
	MPI-20-STD-span-multigrid	**+16.3%**	**42**
	MPI-30-STD-span-multigrid	+16.0%	35

Table 1: Best performing algorithm configurations. The abbreviation in the configuration column denotes *algorithm-m-update scheme-stopping criterion-initialization*. The column *relative runtime* is a measure for whether the configuration works well across different system parameter settings. Specifically it is the configuration runtime relative to the lowest runtime among all configurations and then averaged over all the solved MDPs. The column *#fastest* is the number of MDPs for which the configuration had the lowest runtime.

4.2. CBM discretization

When we constructed the MDP for the CBM case in Section 2.4, we discretized the continuous deterioration process. Let π_D^* denote the policy we obtain from solving the CBM MDP with D discretization intervals. This policy is optimal w.r.t. the discretized deterioration process, and by choosing a large enough D, π_D^* will also be near-optimal w.r.t. the original continuous-state process. In this section, we solve the CBM MDP for increasing values of D and identify the value, where a further increase does not improve the performance of the policy, π_D^*. The CBM MDPs are solved using a discount factor $\gamma = 0.99$. A low tolerance, ϵ, is redundant if we cannot solve the problem for a sufficiently large value of D, hence we use a less strict tolerance $\epsilon = 1$.

Because of the discretization, the value function we get from solving the MDP, $v_{\pi_D^*}$, is only an approximation to the true value of π_D^*. As a way of assessing the true performance of π_D^*, we look

at the total discounted reward obtained from simulating the maintained system, $\mathbf{Y}_t$, when actions are chosen according to $\pi^*_D(\mathbf{Z}_\tau)$. For each value of D and system instance $j = 1, \ldots, 90$, we let $\bar{v}^D_j$ denote a Monte Carlo estimate of the expected total discounted reward when starting with all new components, i.e., in state $\mathbf{Y}_0 = \mathbf{0}$. In Figure 2, $\bar{v}^D_j$ is shown for $N = 4$ and discretization levels $D \in \{2, 3, 4, 6, 8, 12, 16, 24\}$. These particular values of D are used as they appear when starting the multigrid procedure with either $D = 2$ or $D = 3$. We calculate $\bar{v}^D_j$ as the average of 10,000 realizations of 1000 time steps of $\mathbf{Y}_t$. We choose this simulation length because after time 1000 all remaining rewards in an infinitely long horizon account for only $\gamma^{1000}/(1-\gamma) \approx 0.5\%$ of the total.

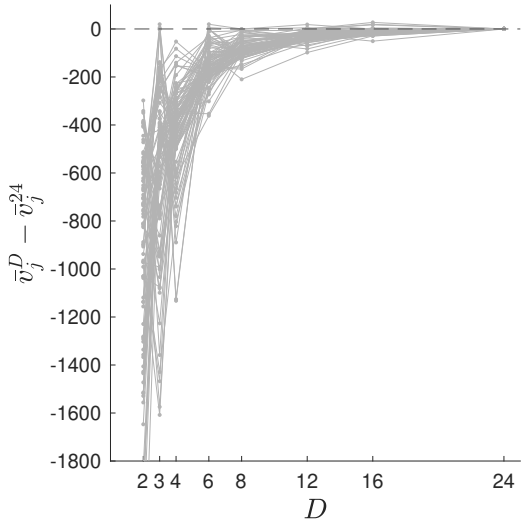


Figure 2: For each 4-component system instance, $j = 1, \ldots, 90$, the estimated expected total discounted reward $\bar{v}^D_j$ is plotted for increasing discretization levels, D. All estimates, $\bar{v}^D_j$ are shifted by $\bar{v}^{24}_j$ in order to show all $j = 1, \ldots, 90$ simultaneously. Standard errors of $\bar{v}^D_j$ lies between 3.4 and 19.6 where the 95th percentile is 11.3.

As Figure 2 indicates, in all the 90 instances of 4-component systems the performance of π^*_D does not improve beyond $D = 16$. A reasonable conjecture is that a higher number of components requires a finer discretization. To test this, we compare the incremental improvement of $\bar{v}^D_j$ for different N. Define $\delta^{D \to D'}$, $D < D'$, as the mean relative increase in the estimated total discounted reward when using policy $\pi^*_{D'}$ instead of policy π^*_D, that is

$$\delta^{D \to D'} = \frac{1}{90} \sum_{j=1}^{90} \frac{\bar{v}^D_j - \bar{v}^{D'}_j}{\bar{v}^D_j}. \tag{11}$$

Table 2 shows the rate at which $\delta^{D\to D'}$ tends towards zero for different values of N. Seeing as $\delta^{12\to 16}$ is well below one percent for $N \leq 4$, the expected total discounted reward from using π_D^* is already close to the asymptote value when $D = 12$. Furthermore, when solving $N = 2$ and $N = 3$ with $D = 64$ we get an estimated $\delta^{12\to 64}$ of 0.48%(0.28%) and 0.75%(0.19%), respectively. Together with the results from Table 2 showing that the incremental improvements decay at the same rate for all values of N up to $D = 12$, this is a strong indication that the performance improvement beyond $D = 12$ is very small.

N	2	3	4	5
$\delta^{2\to 3}$	2.11%(7.77%)	4.82%(8.27%)	5.28%(9.63%)	6.63%(8.37%)
$\delta^{3\to 4}$	2.44%(8.33%)	1.70%(8.06%)	1.67%(8.53%)	1.43%(7.07%)
$\delta^{4\to 6}$	2.63%(3.80%)	4.11%(3.87%)	5.04%(4.46%)	5.10%(3.83%)
$\delta^{6\to 8}$	1.45%(2.27%)	1.18%(1.67%)	1.17%(1.60%)	1.52%(1.45%)
$\delta^{8\to 12}$	0.97%(1.37%)	0.87%(0.78%)	0.97%(0.80%)	1.04%(0.72%)
$\delta^{12\to 16}$	0.15%(0.46%)	0.31%(0.46%)	0.36%(0.32%)	–
$\delta^{16\to 24}$	0.19%(0.25%)	0.23%(0.33%)	0.20%(0.21%)	–

Table 2: The mean relative increase in the estimated total discounted reward for increasing discretization levels. Standard deviations are shown in parentheses. The missing entries take more than one week to calculate.

Olde Keizer et al. (2016) presents a two-component example with $D = 48$, Elwany et al. (2011) uses $D = 20$ for a single-component system, Andriotis & Papakonstantinou (2019) uses $D = 24$ for a 25-component system (solved approximately), and in Sun et al. (2018) the authors suggest using a D such that the interval length is in the same order of magnitude as the precision of the sensors that measure the degradation level. Undoubtedly, the coarsest discretization level for which a near-optimal policy can be obtained depends on which deterioration process is assumed. The gamma process is arguably the most commonly used deterioration process in maintenance optimization literature. As our results indicate, for this deterioration process, setting D lower than in the aforementioned studies is adequate. Indeed, by replicating the gamma process example from Olde Keizer et al. (2016) with different values of D, we find that $D = 9$ results in a policy with the same performance as when $D = 48$.

4.3. Runtime and memory

In this section we solve the models with as many components as possible based on the hardware we use. The TBM MDPs are solved with the MPI-60-STD-span-below configuration and $\epsilon = 0.001$. The CBM MDPs are solved using the MPI-20-STD-span-multigrid configuration with $\epsilon = 1$ and $D = 12$. The results are summarized in Table 3, which shows the runtime and memory usage for the different transition probability storage methods in Section 3.7.

		N			
TBM		4	5	6	7
All θ	store $q(\mathbf{s}'\|\mathbf{s})$	5seconds(3) 21MB(2)	3minutes(1.5) 394MB(58)	1.6hours(0.8) 9GB(2)	– $>$ 128GB
$\theta = 0$	calculate $q(\mathbf{s}'\|\mathbf{s})$	10seconds(5) 8MB(0.4)	7minutes(4) 67MB(10)	4.5hours(2.4) 1GB(0.1)	47hours(2)* 4GB(0.3)*
CBM					
$\theta = 3$	store $q(\mathbf{s}'\|\mathbf{s})$	30seconds(10) 700MB(80)	1hours(0.5) 15GB(1)	– $>$ 128GB	– $>$ 128GB
	store $q(\mathbf{s}'\|\mathbf{0})$	8minutes(2) 8MB(0)	57hours(21) 61MB(0)	$>$ 1week* –	– –
$\theta = 0.2$	store $q(\mathbf{s}'\|\mathbf{s})$	2minutes(0.3) 2GB(0)	9hours(3) 80GB(0.2)	– $>$ 128GB	– $>$ 128GB
	store $q(\mathbf{s}'\|\mathbf{0})$	8minutes(3) 8MB(0)	60hours(19) 61MB(0)	$>$ 1week* –	– –
$\theta = 0$	store $q(\mathbf{s}'\|\mathbf{s})$	2minutes(0.3) 2GB(0.3)	9hours(2) 80GB(0.1)	– $>$ 128GB	– $>$ 128GB
	calculate $q(\mathbf{s}'\|\mathbf{s})$	4minutes(1) 8MB(0)	26hours(4) 55MB(0)	$>$ 1week* –	– –

Table 3: Runtime and used memory for each transition probability storage method from Section 3.7. Each cell is the mean over the 30 different MDPs for either $\theta = 0$, $\theta = 0.2$, or $\theta = 3$, except the first TBM row which includes all 90 MDPs. Parentheses denote standard deviations. (*Parallel implementation using 10 cores.)

In all cases where it is possible to store all nonzero $q(\mathbf{s}'|\mathbf{s})$, $\mathbf{s}', \mathbf{s} \in S$, we can solve the MDP within a relatively short amount of time. The TBM models are faster to solve than the CBM models because the number of nonzero probabilities out of all the $|S|^2$ possible combinations of $\mathbf{s}'$ and $\mathbf{s}$ is smaller in the TBM models, as shown in Table 4. It is worth noting that when storing all

probabilities, the CBM MDPs for which components are strongly dependent ($\theta = 3$) have a lower runtime and use less memory compared to $\theta = 0.2$ and $\theta = 0$. The reason is that deterioration increments, where some components have large jumps and others do not, are very unlikely and therefore estimated to have zero probability. However, the benefit in terms of memory is limited since storing all nonzero $q(\mathbf{s}'|\mathbf{s})$ is still only feasible up to five components.

	N			
TBM	4	5	6	7
$\lvert S\rvert$	5×10^4	9×10^5	1×10^7	2×10^8
Nonzero	1×10^{-4}	2×10^{-5}	2×10^{-6}	4×10^{-7}
CBM				
$\lvert S\rvert$	3×10^4	4×10^5	5×10^6	6×10^7
Nonzero ($\theta < 3$)	0.08	0.05	0.02	–
Nonzero ($\theta = 3$)	0.03	0.01	–	–

Table 4: Mean state space sizes $|S|$ and the mean fraction of transition probabilities that are nonzero, i.e., $|\{(\mathbf{s}', \mathbf{s}) : \mathbf{s}', \mathbf{s} \in S, q(\mathbf{s}', \mathbf{s}) > 0\}|/|S|^2$. The missing entries take more than one week to calculate.

In the case of independent component deterioration ($\theta = 0$), calculating the probabilities at runtime is at most 3 times slower than storing all probabilities. Considering that adding one component generally increases the runtime by a factor of 40 for TBM and 300 for CBM, this is a relatively small difference. Furthermore, the low memory requirements of calculating probabilities at runtime enable us to attempt solving MDPs with more components. Our serial implementation is too slow for $N = 6$ in CBM and $N = 7$ in TBM. Table 3 includes the runtimes of a parallel implementation that divides Equations (7) and (8) between 10 CPU cores when these are calculated for each $\mathbf{s} \in S$. This implementation solves the TBM MDPs with $\theta = 0$ and $N = 6$, in an average of 34 minutes, which is an eightfold speedup compared to the 4.5 hour runtime of the serial version. For the CBM MDPs where $\theta = 0$ and $N = 5$, the 10 cores use 2.6 hours on average, which is a tenfold speedup. Even so, we where not able to solve the $N = 6$ CBM instance within a one-week time limit. It is possible that instances with more components can be solved with more computer resources, given that the parallel algorithm scales linearly up to 10 cores. At some number of components it does, however, become difficult to even store the solution to the optimization problem. For instance, the 128GB available memory allows us to store a value function in an MDP with $|S| = 3.4 \times 10^{10}$ using single-precision floating point numbers. This corresponds

to 9 components for the values of D and D_i we use in our numerical experiments.

Finally we note that our parallel implementation does not scale as well for the CBM MDPs where $\theta = 0.2$ and $\theta = 3$. It solves the $N = 5$ instances in an average of 11 hours, which is only a 5-fold speedup. Combined with the fact that the TBM models, where $\theta > 0$, require a lot of memory, we consider it infeasible to solve larger models with copula dependence between components. There are, however, other ways of modeling stochastic dependence, that allows for efficient runtime calculations of transition probabilities. In the TBM model in Barreto et al. (2014) and the CBM model in Olde Keizer et al. (2018), stochastic dependence is modeled such that the failure rate and deterioration increments are conditionally independent given the current state of the system. Hence, transition probabilities can be written on the form

$$q(\mathbf{s}'|\mathbf{s}) = \prod_{i=1}^{N} q_i(s_i'|\mathbf{s}). \tag{12}$$

This equation resembles Equation (3), and we can calculate transition probabilities at runtime using a procedure similar to the one in Section 3.7.2.

5. Conclusion

In this paper we consider the optimization of component replacements in multi-component systems. We find that other works in maintenance literature assume at least one of the following: discrete system dynamics, single-component system, or a heuristic optimization approach. We present a unified view of TBM and CBM in a general setting where all three of these assumptions are relaxed. We do this by formulating MDP models for TBM and CBM based on the same multi-component system where the component deterioration process is continuous. We use DP to compute optimal policies in the MDPs, which are also optimal with respect to the controlled system when deterioration increments are independent among the components.

For the CBM model, discretization of the continuous deterioration process must be employed in order to use DP. The performance of the resulting policies do not improve when the component condition is discretized into more than twelve levels, which is fewer than suggested in previous studies. According to our results, this number is not sensitive to the system parameters or the number of components. This is an important finding, as it indicates that DP is a feasible solution approach in systems with several components.

We investigate the computational limitations of the proposed TBM and CBM models. An efficient implementation of DP algorithms allows us to solve instances that have 200 million states and 400 thousand states for the TBM and CBM model, respectively. The different size limits are a consequence of the number of possible transitions between states, which is inherently different for the two maintenance approaches. Therefore, multi-component CBM problems are generally harder to solve to optimality than multi-component TBM problems. Furthermore, the limiting factor for the TBM model investigated here is the memory requirement of storing the MDP transition probabilities. However, for the special case of independent component deterioration, the transition probabilities can be calculated efficiently at runtime, allowing for larger instances to be solved via parallelization of the DP algorithm.

Heuristic optimization methods are necessary when dealing with industrial systems composed of dozens of components or assets. However, we argue that the MDPs presented in this study provide a more general and flexible class of policies, and though the optimal policy is difficult to obtain, they serve an important purpose of validating heuristics. Being able to solve large instances to optimality can strengthen statements on how the performance of the heuristics scale with the size of the problem.

Acknowledgments

We would like to express our sincere gratitude to the three anonymous referees for their comments, questions, and suggestions, which have led to major improvements of the content and readability of this paper.

References

Abdel-Hameed, M. (1987). Inspection and maintenance policies of devices subject to deterioration. *Advances in Applied Probability*, *19*, 917–931. doi:10.2307/1427108.

Alaswad, S., & Xiang, Y. (2017). A review on condition-based maintenance optimization models for stochastically deteriorating system. *Reliability Engineering & System Safety*, *157*, 54–63. doi:10.1016/j.ress.2016.08.009.

Andriotis, C. P., & Papakonstantinou, K. G. (2019). Managing engineering systems with large state and action spaces through deep reinforcement learning. *Reliability Engineering & System Safety*, *191*, 106483. doi:10.1016/j.ress.2019.04.036.

Barreto, A. M. S., Pineau, J., & Precup, D. (2014). Policy iteration based on stochastic factorization. *Journal of Artificial Intelligence Research*, *50*, 763–803. doi:10.1613/jair.430.

Boucherie, R. J., & van Dijk, N. M. (Eds.) (2017). *Markov Decision Processes in Practice*. Springer.

Castanier, B., Grall, A., & Bérenguer, C. (2005). A condition-based maintenance policy with non-periodic inspections for a two-unit series system. *Reliability Engineering & System Safety*, *87*, 109–120. doi:10.1016/j.ress.2004.04.013.

Chen, N., Ye, Z.-S., Xiang, Y., & Zhang, L. (2015). Condition-based maintenance using the inverse Gaussian degradation model. *European Journal of Operational Research*, *243*, 190–199. doi:10.1016/j.ejor.2014.11.029.

Chow, C.-S., & Tsitsiklis, J. N. (1991). An optimal one-way multigrid algorithm for discrete-time stochastic control. *IEEE transactions on automatic control*, *36*, 898–914. doi:10.1109/9.133184.

De Jonge, B., & Scarf, P. A. (2020). A review on maintenance optimization. *European Journal of Operational Research*, *285*, 805–824. doi:10.1016/j.ejor.2019.09.047.

Dekker, R., Wildeman, R., & van Egmond, R. (1996). Joint replacement in an operational planning phase. *European Journal of Operational Research*, *91*, 74–88. doi:10.1016/0377-2217(94)00363-7.

Derman, C. (1963). On optimal replacement rules when changes of state are Markovian. In R. Bellman (Ed.), *Mathematical Optimization Techniques* (pp. 201–210). University of California Press Berkeley, CA.

Elwany, A. H., Gebraeel, N. Z., & Maillart, L. M. (2011). Structured replacement policies for components with complex degradation processes and dedicated sensors. *Operations research*, *59*, 684–695. doi:10.1287/opre.1110.0912.

Grothe, O., & Hofert, M. (2015). Construction and sampling of archimedean and nested archimedean Lévy copulas. *Journal of Multivariate Analysis*, *138*, 182–198. doi:10.1016/j.jmva.2014.12.004.

Haurie, A., & l'Ecuyer, P. (1982). A stochastic control approach to group preventive replacement in a multicomponent system. *IEEE Transactions on Automatic Control*, *27*, 387–393. doi:10.1109/TAC.1982.1102919.

Jiang, D. R., & Powell, W. B. (2015). An approximate dynamic programming algorithm for monotone value functions. *Operations Research*, *63*, 1489–1511. doi:10.1287/opre.2015.1425.

Jiang, W., Hong, H., & Ren, J. (2019). Estimation of model parameters of dependent processes constructed using Lévy copulas. *Communications in Statistics - Simulation and Computation*, (pp. 1–17). doi:10.1080/03610918.2019.1565581.

Kurt, M., & Kharoufeh, J. P. (2010). Monotone optimal replacement policies for a Markovian deteriorating system in a controllable environment. *Operations Research Letters*, *38*, 273–279. doi:10.1016/j.orl.2010.03.001.

Li, H., Deloux, E., & Dieulle, L. (2016). A condition-based maintenance policy for multi-component systems with Lévy copulas dependence. *Reliability Engineering & System Safety*, *149*, 44–55. doi:10.1016/j.ress.2015.12.011.

Littman, M. L., Dean, T. L., & Kaelbling, L. P. (1995). On the complexity of Markov decision processes. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI-95), Montreal, Québec, Canada*.

Naderkhani ZG, F., & Makis, V. (2015). Optimal condition-based maintenance policy for a partially observable system with two sampling intervals. *International Journal of Advanced Manufacturing Technology*, *78*, 795–805. doi:10.1007/s00170-014-6651-4.

Neves, M. L., Santiago, L. P., & Maia, C. A. (2011). A condition-based maintenance policy and input parameters estimation for deteriorating systems under periodic inspection. *Computers & Industrial Engineering*, *61*, 503–511.

doi:10.1016/j.cie.2011.04.005.

Nguyen, K. T., Do, P., Huynh, K. T., Bérenguer, C., & Grall, A. (2019). Joint optimization of monitoring quality and replacement decisions in condition-based maintenance. *Reliability Engineering & System Safety*, *189*, 177 – 195. doi:https://doi.org/10.1016/j.ress.2019.04.034.

Nicolai, R. P., & Dekker, R. (2008). Optimal maintenance of multi-component systems: a review. In K. A. H. Kobbacy, & D. P. Murthy (Eds.), *Complex system maintenance handbook* chapter 11. (pp. 263–286). Springer. doi:10.1007/978-1-84800-011-7_11.

Olde Keizer, M. C. A., Flapper, S. D. P., & Teunter, R. H. (2017). Condition-based maintenance policies for systems with multiple dependent components: A review. *European Journal of Operational Research*, *261*, 405–420. doi:https://doi.org/10.1016/j.ejor.2017.02.044.

Olde Keizer, M. C. A., Teunter, R. H., & Veldman, J. (2016). Clustering condition-based maintenance for systems with redundancy and economic dependencies. *European Journal of Operational Research*, *251*, 531–540. doi:10.1016/j.ejor.2015.11.008.

Olde Keizer, M. C. A., Teunter, R. H., Veldman, J., & Babai, M. Z. (2018). Condition-based maintenance for systems with economic dependence and load sharing. *International Journal of Production Economics*, *195*, 319–327.

Özekici, S. (1988). Optimal periodic replacement of multicomponent reliability systems. *Operations Research*, *36*, 542–552. doi:10.1287/opre.36.4.542.

Pineau, J., Gordon, G., & Thrun, S. (2003). Point-based value iteration: An anytime algorithm for pomdps. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence* IJCAI'03 (p. 1025–1030). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Puterman, M. L. (2005). *Markov decision processes: Discrete stochastic dynamic programming*. (2nd ed.). Wiley. doi:10.1002/9780470316887.

Shi, Y., Feng, Q., Shu, Y., & Xiang, Y. (2020). Multi-dimensional Lévy processes with Lévy copulas for multiple dependent degradation processes in lifetime analysis. *Quality Engineering*, *32*, 434–448. doi:10.1080/08982112.2020.1757704.

Sun, Q., Ye, Z.-S., & Chen, N. (2018). Optimal inspection and replacement policies for multi-unit systems subject to degradation. *IEEE Transactions on Reliability*, *67*, 401–413. doi:10.1109/TR.2017.2778283.

Sun, T., Zhao, Q., & Luh, P. B. (2007). Incremental value iteration for time-aggregated Markov-decision processes. *IEEE Transactions on Automatic Control*, *52*, 2177–2182. doi:10.1109/TAC.2007.908359.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. (2nd ed.). MIT press.

van Oosterom, C., Elwany, A., Çelebi, D., & van Houtum, G. (2014). Optimal policies for a delay time model with postponed replacement. *European Journal of Operational Research*, *232*, 186–197. doi:10.1016/j.ejor.2013.06.038.

Vu, H. C., Do, P., Barros, A., & Bérenguer, C. (2014). Maintenance grouping strategy for multi-component systems with dynamic contexts. *Reliability Engineering & System Safety*, *132*, 233–249. doi:10.1016/j.ress.2014.08.002.

Xia, L., Zhao, Q., & Jia, Q.-S. (2008). A structure property of optimal policies for maintenance problems with safety-critical components. *IEEE Transactions on Automation Science and Engineering*, *5*, 519–531. doi:10.1109/TASE.2007.910763.

Xu, J., Liang, Z., Li, Y.-F., & Wang, K. (2021). Generalized condition-based maintenance optimization for multi-component systems considering stochastic dependency and imperfect maintenance. *Reliability Engineering & System Safety*, *211*, 107592. doi:10.1016/j.ress.2021.107592.

Zhang, N., Fouladirad, M., Barros, A., & Zhang, J. (2020). Condition-based maintenance for a k-out-of-n deteriorating system under periodic inspection with failure dependence. *European Journal of Operational Research*, *287*, 159–167. doi:10.1016/j.ejor.2020.04.041.

Appendices

A. Monte Carlo estimation

In both models, we use Algorithm 4.2 from Grothe & Hofert (2015) to sample from the distribution of the one-epoch increments, $\mathbf{X}_1$. The algorithm requires a truncation parameter, K, for the number of jumps to include from an infinite sum of process jumps. For $\theta = 0.2$ we use $K = 4000$ and for $\theta = 3$ we use $K = 30$. These values were determined using the method from Section 5.1 in Grothe & Hofert (2015).

A.1. CBM

When $\theta > 0$, the transition probabilities $q(\mathbf{s}'|\mathbf{s})$, $\mathbf{s}', \mathbf{s} \in S$ are calculated via Equation (10), so we only have to estimate $q(\mathbf{s}|\mathbf{0})$. We do this by sampling M realizations of $\mathbf{X}_1$, denoted $\mathbf{x}_1^k$, $k = 1, \ldots, M$, and then form the estimates as

$$q(\mathbf{s}|\mathbf{0}) = \frac{|\{k : \mathbf{x}_1^k \in \mathbf{I}_\mathbf{s}\}|}{M}, \quad \mathbf{s} \in S. \tag{13}$$

A.2. TBM

When $\theta > 0$, we estimate $q(\mathbf{s}'|\mathbf{s})$ given by Equation (5). This equation is the probability of a specific combination of component failures, conditioned on these components not having failed at the ages given in the vector $\mathbf{s} \in S$. In the following procedure, we therefore account for all the likely combinations of component ages, and for each age combination, then account for each combination of failures.

First, we simulate M trajectories of $\mathbf{X}_\tau$, denoted $\mathbf{x}_\tau^k = (x_\tau^{1,k}, \ldots, x_\tau^{N,k})$, $k = 1, \ldots, M$, $\tau = 0, \ldots, D_{\max}$, where $D_{\max} = \max_i\{D_i\}$. For each k and τ, we then consider all possible combinations of component ages up to age τ, via all the possible replacement times of each component. With the restriction that each component is replaced exactly once, there are $(\tau + 1)^N$ combinations for

N components. Let $\mathbf{r} = (r_1, \ldots, r_N) \in \{0, \ldots, \tau\}^N$ be a vector denoting the replacement times of each component. For each k, τ, and $\mathbf{r} \in \{0, \ldots, \tau\}^N$ we can then construct a realization of $\mathbf{Y}_\tau$ were each component i was last replaced $\tau - r_i$ epochs ago. We define these by

$$\mathbf{y}^k_{\tau,\mathbf{r}} = (y^{1,k}_{\tau,\mathbf{r}}, \ldots, y^{N,k}_{\tau,\mathbf{r}}) = (x^{1,k}_\tau - x^{1,k}_{r_1}, \ldots, x^{N,k}_\tau - x^{N,k}_{r_N}), \tag{14}$$

that is, the k'th simulated trajectory at time τ if we replaced components at the times given in $\mathbf{r} \in \{0, \ldots, \tau\}^N$.

Now, let $V_\mathbf{s}$ be the set of tuples, $(k, \tau, \mathbf{r})$, where $\mathbf{y}^k_{\tau,\mathbf{r}}$ corresponds to the MDP being in state $\mathbf{s} \in S$. Furthermore, we let $W_{\mathbf{s},\mathbf{s}'}$ be the set of tuples, $(k, \tau, \mathbf{r})$, where $\mathbf{y}^k_{\tau,\mathbf{r}}$ and $\mathbf{y}^k_{\tau+1,\mathbf{r}}$ corresponds to the MDP being in state $\mathbf{s} \in S$ at time τ and state $\mathbf{s}' \in S$ at time $\tau + 1$. The estimates $q(\mathbf{s}'|\mathbf{s})$ are now calculated as

$$q(\mathbf{s}'|\mathbf{s}) = \frac{|W_{\mathbf{s},\mathbf{s}'}|}{|V_\mathbf{s}|}, \quad \mathbf{s}, \mathbf{s}' \in S. \tag{15}$$

The sets $V_\mathbf{s}$ and $W_{\mathbf{s},\mathbf{s}'}$ are formally defined as

$$\begin{aligned} V_\mathbf{s} = \big\{ &(k, \tau, \mathbf{r}) : k \in \{1, \ldots, M\}, \tau \in \{0, \ldots, D_{max}\}, \mathbf{r} \in \{0, \ldots, \tau\}^N, \\ &s_i = \tau - r_i \text{ and } y^{i,k}_{\tau,\mathbf{r}} < L \text{ for } s_i < D_i, \\ &y^{i,k}_{\tau,\mathbf{r}} \geq L \text{ for } s_i = D_i \big\}, \quad \mathbf{s} \in S, \end{aligned} \tag{16}$$

and

$$\begin{aligned} W_{\mathbf{s},\mathbf{s}'} = \big\{ &(k, \tau, \mathbf{r}) : (k, \tau, \mathbf{r}) \in V_\mathbf{s}, s'_i = \tau + 1 - r_i \text{ and } y^{i,k}_{\tau+1,\mathbf{r}} < L \text{ for } s'_i < D_i, \\ &y^{i,k}_{\tau+1,\mathbf{r}} \geq L \text{ for } s'_i = D_i \big\}, \quad \mathbf{s}, \mathbf{s}' \in S. \end{aligned} \tag{17}$$

CHAPTER 4

Discretization

Some studies that use MDP for maintenance modeling assume a discrete-state deterioration process (Olde Keizer et al. 2016; Kurt and Kharoufeh 2010; Byon and Ding 2010). This is a logical choice since the theory of DP is mostly concerned with finite or countable state spaces. In Section 2.4 we gave an example of a situation where a continuous-state deterioration process is more suited. However, discretization is required in order to formalize the optimization problem in a way that it can be solved with DP.

Motivated by this, we also used discretization of a continous deterioration process for the CBM model in Chapter 3. In this chapter, we investigate the discretization step for CBM in more detail. More specifically we compare the different ways of going from the continuous process dynamics to the discrete-state transition probabilities for the MDP.

What we present in this chapter is mostly a continuation of the discretization experiment results in Section 4.2 of paper A in Chapter 3. The primary purpose of these experiments is to assess the difficulty of optimizing maintenance for multi-component systems where we have full knowledge about the continuous deterioration process. However, there is also a second interpretation of the experiment, which relates to the practical aspect of maintenance modeling. One motivation for using a discrete-state model is that categorizing the condition of a system into a few qualitative levels leads to a more simplified maintenance decision-making process (Nguyen et al. 2019). But if too few discrete levels are used, some performance of the resulting maintenance policy is undoubtedly sacrificed. The results we present in the following paper also illustrate how much performance is lost if the true deterioration of the system follows a continuous-state process, but we as decision-makers only model the deterioration with a finite number of qualitative levels.

4.1 Independent components

In the following paper, we only change the model from Chapter 3 in a few ways. Most importantly, we now assume that the deterioration processes of the components, X_t^i, $i = 1, \ldots, N$, are independent. The reason is that some of the methods of discretization we test, require that we know the distribution function of the deterioration increments. In Chapter 3, we used a copula function to model stochastic dependence between components, which resulted in the distribution of the joint process, $\mathbf{X}_t$, not

having an analytical expression. The discretization method we used in Chapter 3, which we in the following refer to as *exact integration*, was therefore estimated with Monte Carlo simulation. Among all the studies that use discretization for the purpose of MDP modeling, exact integration is the most commonly used method. It turns out that there is a better method that produces a more accurate approximation to the continuous deterioration process.

4.2 Unichain condition

The most common utility in maintenance optimization papers is the cost rate, i.e. long-run average reward per time unit in MDP terminology. In Chapter 3, we used another utility, namely the expected total discounted reward. The reason for this is that the DP algorithms using Gauss-Seidel or Successive over-relaxation iterations cannot be used with the cost rate utility. In the present chapter we do, however, consider the cost rate, but in order to guarantee the convergence of standard DP algorithms under this optimality criterion, a *unichain* condition must be satisfied: for every policy, $\pi : S \to A$, the corresponding transition probability matrix, $P_\pi = [p(j|i, \pi(i))]_{ij}$, $i, j \in S$, must consist of a single recurrent class plus a possibly empty set of transient states (Puterman 2005, p. 348).

In all the models we consider, we use the gamma process as the underlying true deterioration process. The increments of this process between two decision epochs are gamma distributed, and this distribution has support $(0, \infty)$, hence there is positive probability that all components will exceed the failure limit, L, and fail. In the MPDs we construct via discretization, the state space is on the form $S = \{0, \dots, D\}^N$, where D is the failed state of a component. Because all components can fail between any two epochs, we get $p(s'|s, a) > 0$, where $s' = (D, \dots, D)$ for all $s \in S$ and $a \in A$. Therefore, P_π can only contain one recurrent class and the action $\pi(s')$ decides the size of it.

4.3 Paper B

Discretization of continuous deterioration processes for multi-component condition-based maintenance optimization*

Jesper Fink Andersen[a,*], Bo Friis Nielsen[a]

[a] *Technical University of Denmark, Department of Applied Mathematics and Computer Science, Anker Engelunds Vej 1, 2800 Kgs. Lyngby, Denmark*

Abstract

In this paper we take a closer look at the issue of discretization that arises when combining continuous deterioration processes with dynamic programming. We consider an example of a multi-component condition-based maintenance problem modeled as a Markov decision process. Components are assumed to deteriorate independently according to a gamma process. We compare the different methods of discretizing this process that have been tried in previous studies. We identify a discretization scheme that accurately estimates the true value of a policy with respect to the continuous deterioration process, and we present an exact evaluation of this scheme, which has so far only been estimated with Monte Carlo methods. Furthermore, we show how a more accurate discretization enables us to solve the optimization problem faster in certain cases using a multigrid algorithm.

Keywords: Discretization, Maintenance, Multi-component system, Markov decision process, Dynamic programming

1. Introduction

In this paper we consider condition-based maintenance optimization in multi-component systems modeled using Markov Decision Processes (MDPs). An MDP is a versatile framework capable of modeling a wide variety of maintenance scenarios. Another common maintenance optimization approach is cost minimization based on renewal reward theory [1]. Whereas the renewal theory optimization technique requires that we impose a certain parametric structure, for instance a control limit structure, the space of considered policies when solving an MDP is not limited by any

*The funding for this work was provided by the Technical University of Denmark
*Corresponding author
Email addresses: jfan@dtu.dk (Jesper Fink Andersen), bfni@dtu.dk (Bo Friis Nielsen)

such restrictions. The limitation of using MDPs is that the dynamic programming algorithms used for finding the optimal policy require that the state space of the MDP is discrete. In a CBM context this means the deterioration of components must be modeled as a discrete-time Markov chain. The many transition probabilities in this model can, however, be difficult to specify, and the states only represent a categorization into high-level condition states such as "new", "mild deterioration", "severe deterioration", and "failed", see e.g. [2, 3, 4, 5]. There are many systems, where the component deterioration is most naturally modeled as a continuous-time and continuous-state stochastic process. For instance, this can be the case if condition monitoring is based on measurements of a continuous variable, such as vibration magnitude, product scrap rate, temperature, or corrosion penetration depth. A prevalent example of a continuous deterioration process is the gamma process [6], which we also use as an example in this paper. If there is not enough data to directly estimate all transition probabilities in a discrete state transition matrix, fitting a continuous deterioration process is also a way of interpolating in the areas of the state space for which we have little data.

There is a benefit in combining the dynamic programming with continuous deterioration processes, since the former enables us to identify the complex structures of optimal policies in multi-component systems, and using the latter can be a more accurate model of the system dynamics. Under an additional and common assumption that the system is periodically inspected, the resulting optimization problem becomes an MDP with an uncountable state space. General algorithms for solving these MDPs are analyzed in [7] and [8], and in both cases the solution approach is to approximate the optimal policy by discretizing the continuous dynamics and solving a discrete-state version of the problem. In this study, we consider an uncountable state space, $[0, L)^N \subset \mathbb{R}^N$, $L > 0$, $N \in \mathbb{N}$ and perform the discretization by dividing this state space into equally sized regions. From here on we refer to the definition of the transition probabilities between regions as a discretization scheme or method.

In [9], the author analyzes a periodic replacement problem in a multi-component system with a continuous deterioration process, similar to the example problem we consider in this paper. A characterization of the optimal policy is obtained, but even so the conclusion is that dynamic programming is still needed to compute the policy, hence the deterioration process must be dicsretized.

This practice has also been adopted in other maintenance optimization studies, but the issue of

discretization is usually treated superficially. In single-component systems considered in [10] and [**?**], the choice of the discretization method is not a large concern, since the problem can be solved easily with a very fine discretization. For multi-component models that are solved with dynamic programming [11, 12, 13, 14], the accuracy of the discretized deterioration process matters because a small increase in the number of discretization intervals causes a large increase in the total number of states in the MDP. As we demonstrate in this paper, the accuracy of the different discretization methods used in [11, 12, 13, 14] is not the same. This has some interesting consequences for how close the resulting policy is to being optimal, and how fast we can solve the MDP with a desired discretization fineness. To the best of our knowledge, a comparison of different discretization schemes has not been carried out before, and therefore the results we present are novel.

We find that the most accurate method of discretization is the one used in [11] and [15]. The uncountable state space is divided into equally sized intervals and the frequency of transitions from one interval to another is estimated with Monte Carlo simulation. We show how to numerically evaluate these probabilities for the continuous deterioration process we use in our example, which eliminates the estimation error. In [15] this discretization method is initially presented as a way of estimating the transition probabilities in the discretized process from data. In this light, the numerical experiments we present are also a validation of the practice of defining the component deterioration states as a small number of qualitative levels for instance by expert judgement. We simulate the maintained system with the continuous deterioration process and choose maintenance actions according to the policies obtained from an MDP with a varying number of discretization intervals. From the maintenance costs we observe in the simulations, we get an indication of how effective this method of maintenance optimization is. Furthermore, we show how a better discretization method can lead to a faster reduction in the approximation error, and in some cases this allows us to solve the problem faster via a multigrid algorithm similar to that analyzed in [7].

The rest of the paper is structured as follows. In Section 2 we describe the system assumptions and the finite-state MDP formulation of the CBM model. In Section 3 we briefly summarize four different discretization methods and derive an exact evaluation procedure for one of them. In Section 4 we summarize the dynamic programming algorithms we use to solve the MDP. In Section 5 we present the empirical results of the comparison on accuracy and algorithm runtime for the different discretization methods, and in Section 6 we provide a conclusion to the study.

2. Model formulation

As a basis for comparison of the different discretization methods, we consider a multi-component replacement optimization problem, with an infinite planning horizon, and where the optimal policy has a complex structure.

The system consists of N components. Each component $i = 1, \ldots, N$ deteriorate independently of the other components according to a time-homogeneous gamma process, $\{X_t^i\}_{t \geq 0}$, i.e., we assume $X_0^i = 0$ and for $0 \leq s < t$, $X_t^i - X_s^i \sim \text{Gamma}(\alpha_i(t-s), \beta_i)$, where α_i is the shape parameter and β_i is the rate parameter of component i. Component i fails when X_t^i reaches a failure threshold L.

We assume the system is periodically inspected at discrete points in time, $\tau \in \mathbb{N}_0 = \{0, 1, \ldots\}$, also called decision epochs in MDP terminology. At each inspection we can decide to replace any combination of components. Replacements are assumed to be instantaneous, and bring the components back to an as-good-as-new state. The cost of a preventive replacement, before failure, is c_p^i. If the replacement happens after a failure, we incur a larger corrective replacement cost, c_c^i. Furthermore, we assume there is a joint setup cost, c_s, if at least one component is replaced. The system is assumed to have a K-out-of-N reliability structure, meaning that the system is functioning as long as K components are functioning. We therefore incur a failure cost, c_f, if less than K components are functioning at any given inspection.

2.1. Finite MDP formulation

We formulate the optimization problem as an finite MDP in order to optimize the replacement policy with dynamic programming. At each decision epoch, $\tau \in \mathbb{N}_0$, the following events take place in the MDP: The system occupies a state $\mathbf{s} \in S$, where S is a finite set; an action, $\mathbf{a}$, is chosen from a finite set of actions, A; we receive a reward $r(\mathbf{s}, \mathbf{a})$; and the system transitions to another state $\mathbf{s}' \in S$ with probability $p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$.

The system state is a vector $\mathbf{s} = (s_1, \ldots, s_n) \in S$ where $s_i \in S_D = \{0, \ldots, D\}$ is the state of component i and $S = S_D^N$. The discrete set, S_D, is formed by dividing the interval $[0, L)$ into D equally sized intervals $I_k = [kL/D, (k+1)L/D)$, $k = 0, \ldots, D-1$, and we let $I_D = [L, \infty)$. Component i being in the discrete state $s_i \in S_D$ at epoch τ then corresponds to $X_\tau^i \in I_{s_i}$.

Replacement actions are chosen from the set $A = \{0, 1\}^N$. An element a_i of an action $\mathbf{a} = (a_1, \ldots, a_N) \in A$ equals 1 (0) if we replace (do not replace) component i.

The reward function is defined as

$$r(\mathbf{s},\mathbf{a}) = \sum_{i=1}^{N} a_i \left(c_p^i \mathbf{1}_{s_i<D} + c_c^i \mathbf{1}_{s_i=D}\right) + c_s \left(1 - \prod_{i=1}^{N}(1-a_i)\right) + c_f \mathbf{1}_{|\{i:s_i<D\}|<K}, \tag{1}$$

where $\mathbf{1}_A$ is the indicator function for event A. The first term of Equation (1) is the replacement costs. The second and third terms account for the setup cost and system failure cost, respectively. We formulate the MDP in terms of maximizing rewards, but we note that c_p^i, c_c^i, c_s, and c_f are all negative numbers, so $r(\mathbf{s},\mathbf{a}) < 0$ for all $\mathbf{s} \in S$, and $\mathbf{a} \in A$.

Since the replacement of a component is assumed to be instantaneous, the transition probabilities, $p(\mathbf{s}'|\mathbf{s},\mathbf{a})$, can be described in a convenient way. Replacement action $\mathbf{a}$ instantly moves the system from state $\mathbf{s}$ to a *post-decision* state that has zero entries for the replaced components, $((1-a_1)s_1,\ldots,(1-a_N)s_N) \in S$. It therefore suffices to specify the transition probability from any post-decision state $\mathbf{s} \in S$ to $\mathbf{s}' \in S$, which we denote $q(\mathbf{s}'|\mathbf{s})$. Furthermore, since components deteriorate independently, $q(\mathbf{s}'|\mathbf{s})$ can be written as a product

$$q(\mathbf{s}'|\mathbf{s}) = \prod_{i=1}^{N} q_i(s_i'|s_i), \tag{2}$$

where $q_i(s_i'|s_i)$ is the probability of component i transitioning from post-decision state $s_i \in S_i$ to state $s_i' \in S_i$.

3. Discretization methods

In this section we describe the different ways of constructing the MDP component transition probabilities, $q_i(s'|s)$, $s, s' \in S_i$, $i = 1,\ldots,N$ from the evolution of the continuous deterioration process, X_t^i. In principle, the best choice is

$$P(X_{\tau+1}^i \in I_{s'} | X_\tau^i \in I_s), \tag{3}$$

however, this probability is dependent on the history of occupied states at epochs, $0,\ldots,\tau-1$. For instance, knowing that $X_{\tau-2}^i, X_{\tau-1}^i \in I_s$ provides information about the exact location of X_τ^i in the interval I_s, thus influencing the probability of $X_{\tau+1}^i$ being in interval $I_{s'}$. In an infinite-horizon MDP, $q_i(s'|s)$ must be independent of time and the policy being used, and any definition that obeys this therefore ends up being an approximation to Equation (3). There are several choices for the construction of $q_i(s'|s)$ for which the approximation error will diminish, as the number of

discretization intervals, D, increases. In the remainder of this section we describe four different methods.

Since the underlying deterioration process is non-negative and has stationary increments, it is convenient in three of the methods to define $q_i(s'|s)$ from the probability of component i advancing k discrete states, which we denote $u_{i,k}$, $k = 0, \ldots, D-1$. We then have

$$q_i(s'|s) = \begin{cases} u_{i,s'-s} & s \leq s' < D \\ 1 - \sum_{k=0}^{D-s-1} u_{i,k} & s \leq s' = D \\ 0 & s' < s. \end{cases} \tag{4}$$

We let $f_{X_1^i}$, and $F_{X_1^i}$ denote the probability density function and the distribution function of the one-period increment of component i, respectively.

3.1. Normalized density

This method is in used in [7**?** , 16]. For the problem we consider, the method defines the probability of advancing k discrete states as a normalized ratio of the density, $f_{X_1^i}$. The width of one interval is L/D and $f_{X_1^i}(kL/D)$ is therefore the density at the point where X_t^i advances exactly k intervals. We use this to define $u_{i,k}$ as

$$u_{i,k} = \frac{f_{X_1^i}(kL/D)}{\sum_{j=0}^{\infty} f_{X_1^i}(jL/D)}, \quad k = 0, \ldots, D-1. \tag{5}$$

The sum in the denominator is truncated at an appropriately large value depending on the gamma process parameters, α_i and β_i in the density function, $f_{X_1^i}$. We let q_i^{den} denote the transition probabilities we obtain by using $u_{i,k}$ from Equation (5) in Equation (4). This method is the most general among those we consider in this paper. The reason is that it does not require the discrete states to be defined from a partition of the uncountable state space into disjoint regions, like the intervals, I_k, in Section 2.1. Instead, the discretization can simply be a finite set of, possibly random, sample points in the uncountable state space [16].

3.2. Exact integration

The probabilities $q_i^{\text{den}}(s'|s)$ can be seen as the probability of X_t^i transitioning from one representative point in I_s to another representative point in $I_{s'}$. In this next method, we construct $u_{i,k}$ as the probability of X_t^i transitioning from one representative point in I_s to any point in I_{s+k}. The name of the method is adopted from [16]. In [16] and [12] the representative point of interval I_s is

the midpoint, $(s+1/2)L/D$, while [10] uses the right endpoint $(s+1)L/D$. In Section 5 we only report the results from using the midpoint, as we found this to be the best choice for the example problem. Using the distance from the midpoint of I_s to the endpoints of I_{s+k} we get

$$u_{i,k} = F_{X_1^i}((k+1/2)L/D) - F_{X_1^i}((k-1/2)L/D), \quad k = 0, \ldots, D-1. \tag{6}$$

We let q_i^{int} denote the transition probabilities we obtain by using $u_{i,k}$ from Equation (6) in Equation (4).

3.3. Uniform in origin interval

In [17], a slightly more advanced discretization scheme is proposed. Instead of using a representative point in I_s, an approximation of the probability in Equation (3) is formed by assuming X_τ^i is uniformly distributed in I_s. This leads to

$$u_{i,k} = \int_0^1 F_{X_1^i}((k+1-x)L/D) - F_{X_1^i}((k-x)L/D)dx, \quad k = 0, \ldots, D-1. \tag{7}$$

We use q_i^{uni} to denote the transition probabilities we obtain by using $u_{i,k}$ from Equation (7) in Equation (4). It is argued in [17] that the uniform distribution is a good approximation to the true distribution of the deterioration level within an interval I_s in all but the first intervals, that is, when s in $q_i^{\text{uni}}(s'|s)$ is a low number.

3.4. Expected number of transitions

When we discretize, we lose some information about the original continuous deterioration process since $P(X_{\tau+1}^i \in I_{s'} | X_\tau^i \in X_s)$ is dependent on τ. The density normalization, q_i^{den}, and exact integration, q_i^{int}, do not account for this time dependency, while the uniform method, q_i^{uni} adjusts for it. However, a replacement is assumed to bring a component back to the as-good-as-new state, so it is obvious how the uniform method is inaccurate at least in the first epoch after a replacement, since X_0^i equals zero and is not uniformly distributed in $I_0 = [0, L/D)$. There is a way of lessening this discrepancy and averaging out the time dependency. The idea is to form estimates $q_i(s'|s)$ from the expected number of times that the process X_τ^i, $\tau = \mathbb{N}_0$, moves from I_s to $I_{s'}$ before it exceeds the failure limit, L. This method is also mentioned in [15] as a way of estimating $q_i(s'|s)$ directly from field data. The discrete transition probabilities are here calculated as

$$q_i(s'|s) = \frac{\text{Number of transitions from } I_s \text{ to } I_{s'}}{\text{Number of data in } I_s}. \tag{8}$$

When discretizing the continuous deterioration process, we could use Monte Carlo simulation of X_t^i to generate a data set and form estimates from Equation (8), which is also the approach used in [15] and [11]. The estimates in Equation (8) can also be written as an analytical expression, and because we know $F_{X_t^i}$, this can be evaluated numerically rather than via simulation. We observe X_t^i at the decision epochs $\tau = \mathbb{N}_0$. Let B_s, $0 \leq s < D$, denote the number of times X_τ^i is found in interval I_s, that is,

$$B_s = |\{\tau \in \mathbb{N}_0 : X_\tau^i \in I_s\}|.$$

Let $C_{s,s'}$, $0 \leq s < D$ and $0 \leq s' \leq D$ denote the number of times X_t^i transitions from I_s to $I_{s'}$ from one epoch to the next,

$$C_{s,s'} = |\{\tau \in \mathbb{N}_0 : X_\tau^i \in I_s, X_{\tau+1}^i \in I_{s'}\}|.$$

The transition probabilities, which we name $q_i^{\exp}$, are then defined as

$$q_i^{\exp}(s'|s) = \frac{E[C_{s,s'}]}{E[B_s]} = \frac{\sum_{\tau=0}^{\infty} P(X_\tau^i \in I_s, X_{\tau+1}^i \in I_{s'})}{\sum_{\tau=0}^{\infty} P(X_\tau^i \in I_s)}, \tag{9}$$

for $0 \leq s < D, 0 \leq s' \leq D$ and $q_i^{\exp}(D|D) = 1$. The summands in the denominator in Equation (9) are calculated by

$$P(X_\tau^i \in I_s) = F_{X_\tau^i}((s+1)L/D) - F_{X_\tau^i}(sL/D)$$

and the summands in the numerator by

$$\begin{aligned}
&P(X_\tau^i \in I_s, X_{\tau+1}^i \in I_{s'}) \\
&= \int_{sL/D}^{(s+1)L/D} P(X_{\tau+1}^i \in I_{s'} | X_\tau^i = x) f_{X_\tau^i}(x) dx \\
&= \int_{sL/D}^{(s+1)L/D} P(s'L/D - x \leq X_{\tau+1}^i - X_\tau^i < (s'+1)L/D - x) f_{X_\tau^i}(x) dx \\
&= \int_{sL/D}^{(s+1)L/D} \left[F_{X_1^i}((s'+1)L/D - x) - F_{X_1^i}(s'L/D - x) \right] f_{X_\tau^i}(x) dx
\end{aligned}$$

for $0 \leq s \leq s' < D$ and

$$P(X_\tau^i \in I_s, X_{\tau+1}^i \in I_D) = \int_{sL/D}^{(s+1)L/D} \left[1 - F_{X_1^i}(L - x) \right] f_{X_\tau^i}(x) dx, \quad s < D. \tag{10}$$

We use Gauss-Kronrod quadrature to evaluate the integrals numerically. The summands in the expectations in Equation (9) quickly tend to 0 as $\tau \to \infty$ for the parameters α_i and β_i of the

gamma density function, $f_{X_t^i}$, we use in Section 5. Therefore, we can truncate the sums at an appropriately large number of terms. In Section 5, we truncated at 30 terms.

The method presented in this section is only applicable when components deteriorate independently. In [14] and [12] stochastic dependency among components is modeled using copula functions. Since the resulting processes have no analytical expressions for the joint distribution of deterioration increments, these studies instead use Monte Carlo simulation to estimate the exact integration method from Section 3.2.

4. Dynamic programming

In this section, we briefly review the Value Iteration (VI) algorithm we use to find an optimal policy for the MDP defined by S, A , $p(\cdot|\cdot,\cdot)$, and $r(\cdot,\cdot)$ in Section 2.1. In the set of all mappings from S to A, Π, we seek to find a policy, $\pi \in \Pi$, that maximizes the average reward per period, which is often called the cost rate in maintenance literature. The VI algorithm solves this problem by successively computing for $n \in \mathbb{N}_0$

$$v_{n+1}(\mathbf{s}) = \max_{\mathbf{a}\in A}\left\{ r(\mathbf{s},\mathbf{a}) + \sum_{\mathbf{s}'\in S} p(\mathbf{s}'|\mathbf{s},\mathbf{a}) v_n(\mathbf{s}') \right\} \quad \forall \mathbf{s} \in S, \tag{11}$$

where $v_n : S \to \mathbb{R}$ is the value function. The iterative process is stopped when

$$\max_{\mathbf{s}\in S}\{v_{n+1}(\mathbf{s}) - v_n(\mathbf{s})\} - \min_{\mathbf{s}\in S}\{v_{n+1}(\mathbf{s}) - v_n(\mathbf{s})\} < \epsilon, \tag{12}$$

where $\epsilon > 0$. When this criterion is met, an approximation to the optimal average reward per period is given by

$$g_D = (\max_{\mathbf{s}\in S}\{v_{n+1}(\mathbf{s}) - v_n(\mathbf{s})\} + \min_{\mathbf{s}\in S}\{v_{n+1}(\mathbf{s}) - v_n(\mathbf{s})\})/2, \tag{13}$$

which is at most $\epsilon/2$ from the true value [**?**]. The final policy, π_D, consists of the maximizing actions in the final calculation of Equation (11), and we use v_D to denote the final value function. The subscript in g_D, π_D, and v_D indicates this is the solution when solving the MDP from Section 2.1 with D discretization intervals. In addition, we let π_D^{den}, π_D^{int}, π_D^{uni}, and π_D^{exp} denote the policies and g_D^{den}, g_D^{int}, g_D^{uni}, and g_D^{exp} denote the associated cost rate for the MDP with component transition probability definitions indicated by the superscript. Each of these policies are optimal with respect to a system that obeys a discretized version of the deterioration dynamics. For a sufficiently large

value of D, they are also near-optimal for the system with continuous deterioration. We use g^* and π^* to denote the solution to the continuous-state MDP formulation of the problem. We cannot compute g^* and π^* exactly, but these are the asymptotic values that g_D and π_D approach as D increases.

We also consider the Gauss-Seidel version of VI, where the optimality criterion is the expected total discounted reward. The update equation is here given by

$$v_{n+1}(\mathbf{s}) = \max_{\mathbf{a}\in A}\left\{ r(\mathbf{s},\mathbf{a}) + \gamma\left[\sum_{\mathbf{s}'<\mathbf{s}} p(\mathbf{s}'|\mathbf{s},\mathbf{a})v_{n+1}(\mathbf{s}') + \sum_{\mathbf{s}'\geq\mathbf{s}} p(\mathbf{s}'|\mathbf{s},\mathbf{a})v_n(\mathbf{s}')\right]\right\} \quad \forall \mathbf{s}\in S, \tag{14}$$

where $0 \leq \gamma < 1$ is the discount factor. This version of VI is terminated when

$$\max_{\mathbf{s}\in S}\{v_{n+1}(\mathbf{s}) - v_n(\mathbf{s})\} < \epsilon(1-\gamma)/2\gamma, \tag{15}$$

at which point $v_{n+1}(\mathbf{s})$ is within $\epsilon/2$ from the optimal value function for all $\mathbf{s}\in S$.

5. Results

In this section, we compare the different discretization methods by solving the multi-component replacement problem with parameters, $N = 2$, $K = 1$, $L = 1$, $c_s = -30$, $c_f = -1000$, $\alpha_1 = 1.67$, $\beta_1 = 7.27$, $c_p^1 = -33.43$, $c_c^1 = -54.04$, $\alpha_2 = 1.78$, $\beta_2 = 6.88$, $c_p^2 = -16.24$, and $c_c^2 = -52.19$. The numerical experiments we present have been carried out on 10 parameter settings for two components, and 10 settings for four components. The results from the different settings all exhibited the same patterns, so we only present the results from one of the parameter settings.

5.1. Cost rate comparison

As a way of assessing how close π_D is to the globally optimal policy, π^*, we look at the cost rate obtained from simulating the system with the continuous deterioration process and choosing actions according to π_D. At each epoch in the simulation, the continuous state of each component will belong to one of the intervals I_s, $s = 0,\ldots,D$, and the replacement action is determined by $\pi_D(s)$. This can be seen as a piece-wise flat extension of π_D to the uncountable state space, $[0,)^N$. For each value of D, we let $\bar{g}_D$ denote the estimated cost rate, which we calculate from the average of 10^8 time steps of the maintained system, disregarding the rewards from the first 10^4 steps to avoid any influence of the transient phase induced by the initial state. Figure 1 shows, g_D and $\bar{g}_D$

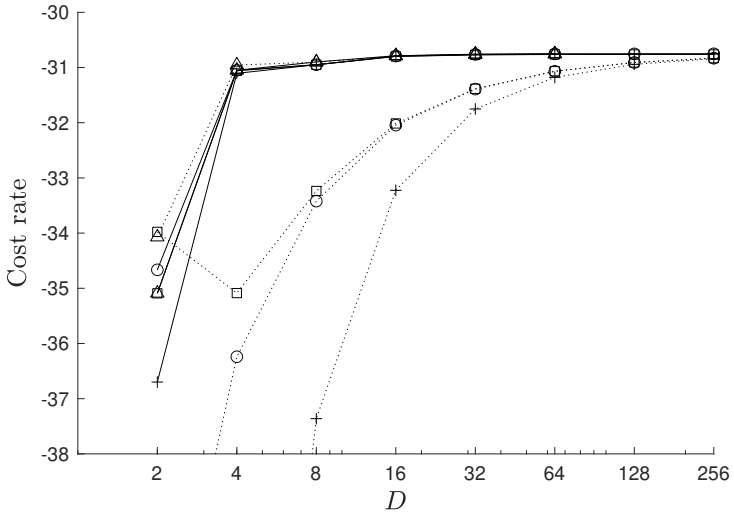


Figure 1: The cost rate for different number of discretization intervals, D. Cost rates from MDP solution, g_D (dotted) and estimates, $\bar{g}_D$, obtained from simulation of the continuous deterioration (solid). The discrete state transition probabilities are: density normalization (plus sign), exact integration (square), uniform in origin interval (circle), and expected number of transitions (triangle).

for different values of D, when the transition probabilities in the MDP are calculated with either q_i^{den}, q_i^{int}, q_i^{uni}, or q_i^{exp}. All methods produce good policies already at $D = 4$, even though g_D^{exp} is the only MDP-estimated cost rate close to the realized cost rate $\bar{g}_D^{\text{exp}}$, indicating that q_i^{exp} is a better approximation of the continuous deterioration process. As shown in Figure 2, already at $D = 4$ the policies π_D^{den}, π_D^{int}, π_D^{uni}, and π_D^{exp} prescribe the same replacement actions on most of the state space. Furthermore, from the structure of the $D = 16$ policies, it appears that the asymptotic optimal policy, π^*, consists of few but irregularly shaped regions. Even for low values of D, the policies π_D^{den}, π_D^{int}, π_D^{uni}, and π_D^{exp} only differ close to the boundary of these regions. For $D = 4$ the transition matrices for the first component in the four different methods are

$$[q_1^{\text{den}}]_{s,s'} = \begin{pmatrix} 0.0000 & 0.7540 & 0.1945 & 0.0414 & 0.0100 \\ 0.0000 & 0.0000 & 0.7540 & 0.1945 & 0.0514 \\ 0.0000 & 0.0000 & 0.0000 & 0.7540 & 0.2460 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{pmatrix}. \tag{16}$$

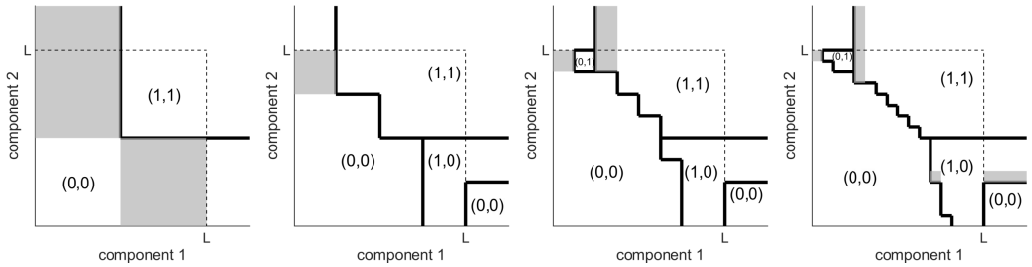


Figure 2: Policies π_D^{den}, π_D^{int}, π_D^{uni}, and π_D^{exp} superimposed on each other, with gray areas indicating where they are not identical. From left to right the panels show, $D = 2$, $D = 4$, $D = 8$, $D = 16$.

$$[q_1^{\text{int}}]_{s,s'} = \begin{pmatrix} 0.3295 & 0.4972 & 0.1365 & 0.0296 & 0.0072 \\ 0.0000 & 0.3295 & 0.4972 & 0.1365 & 0.0368 \\ 0.0000 & 0.0000 & 0.3295 & 0.4972 & 0.1733 \\ 0.0000 & 0.0000 & 0.0000 & 0.3295 & 0.6705 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{pmatrix}, \tag{17}$$

$$[q_1^{\text{uni}}]_{s,s'} = \begin{pmatrix} 0.3212 & 0.4907 & 0.1474 & 0.0327 & 0.0081 \\ 0.0000 & 0.3212 & 0.4907 & 0.1474 & 0.0407 \\ 0.0000 & 0.0000 & 0.3212 & 0.4907 & 0.1881 \\ 0.0000 & 0.0000 & 0.0000 & 0.3212 & 0.6788 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{pmatrix}, \tag{18}$$

and

$$[q_1^{\text{exp}}]_{s,s'} = \begin{pmatrix} 0.4721 & 0.3892 & 0.1091 & 0.0237 & 0.0058 \\ 0.0000 & 0.3205 & 0.4911 & 0.1476 & 0.0408 \\ 0.0000 & 0.0000 & 0.3212 & 0.4907 & 0.1882 \\ 0.0000 & 0.0000 & 0.0000 & 0.3212 & 0.6788 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{pmatrix}. \tag{19}$$

The fact that q_1^{den} deviates substantially from the other three methods and still results in the four policies for $D = 4$ being identical in all but one state, demonstrates a degree of robustness to transition probabilities being inaccurate.

In terms of the accuracy of the cost rate estimate from the MDP solution, g_D, the expected transitions method, q_i^{exp} is the only method that results in a consistently accurate estimate. Surprisingly, the uniform method does not produce a more accurate cost rate estimate than the simpler

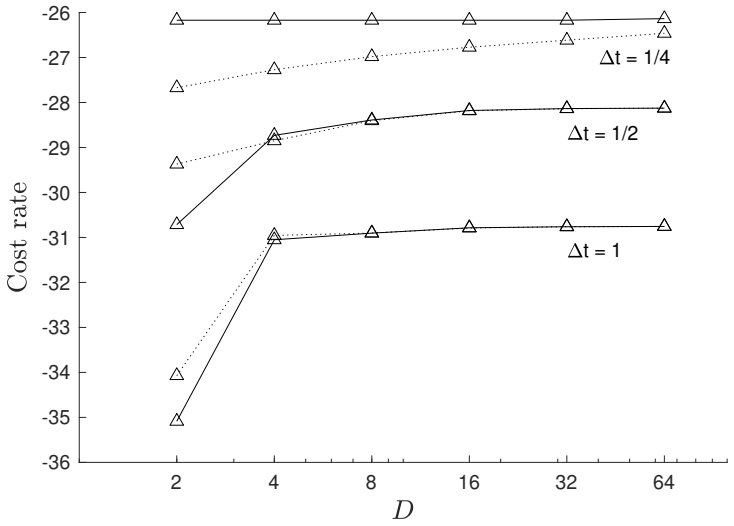


Figure 3: The estimated cost rate g_D^{exp} (dotted) and the realized cost rate $\bar{g}_D^{\text{exp}}$ (solid) for different inspection interval lengths, Δt.

exact integration method. The transition probabilities from the uniform method, q_1^{uni}, are quite close in value to the expected transitions method, q_1^{exp}, in all but the first row. Evidently, this difference has a quite large influence on the cost rate estimate, g_D^{uni}. As it is much faster to calculate q_i^{uni} than q_i^{exp} we tried improving the uniform method by splitting the interval $I_0 = [0, L/D)$ into two discrete states, such that $S_D = \Delta \cup \{0, \ldots, D\}$, where state $\Delta \in S_D$ corresponds to $X_\tau^i = 0$ and letting state $0 \in S_D$ correspond to $X_\tau^i \in (0, L/D)$. The transitions from Δ to the other states $s' \in \{0, \ldots, D-1\}$ are calculated as $F_{X_1^i}((s'+1)L/D) - F_{X_1^i}(s'L/D)$ and transition to state D is with probability $1 - F_{X_1^i}(L)$. Unfortunately this only improves the estimate g_D to halfway between g_D^{uni} and g_D^{exp}. Alternatively, the two methods can be combined such that $q_i^{\text{exp}}(s'|s)$ is used for the first few values of s and $q_i^{\text{uni}}(s'|s)$ is used for the larger values of s, which saves a lot of computation when D is large.

In Section 2 we defined the model such that inspections occur with unit time intervals. For the considered example parameters, the number of inspections in between replacements is 2.8 on average when following the π_D^{exp} policies. This may explain why only $D = 4$ discretization intervals are needed to get a near-optimal policy. Let Δt denote the time between inspections. Figure 3 shows the cost rates g_D^{exp} and $\bar{g}_D^{\text{exp}}$ for different values of Δt. An increased frequency of inspections

allows for components to deteriorate closer to the failure threshold before being replaced, which explains the improved cost rate for $\Delta t = 1/2$ and $\Delta t = 1/4$. When $\Delta t = 1/2$ we need more than $D = 4$ intervals to get the full benefit of these just-in-time replacements, which is shown in Figure 3 by the slightly steeper increase between $D = 4$ and $D = 16$ for $\Delta t = 1/2$ compared to $\Delta t = 1$. When $\Delta t = 1/4$ the simulated cost rate, $\bar{g}_D^{\text{exp}}$, is constant across all values of D. Looking at the corresponding policies, π_D^{exp}, this is because component one is never replaced once it has failed, and coincidentally component two should be replaced approximately when $X_t^2 > 1/2$. This policy structure can be captured already at $D = 2$. We consider a 1-out-of-2 system and when it is inspected very often, the cost of keeping both components working outweighs the cost of a rare system failure. This behaviour is quite specific to the particular system parameter setting we have chosen, but it illustrates an extreme case where the cost rate estimate, g_D^{exp}, is inaccurate. Even so, g_D^{exp} is still the most accurate in both the $\Delta t = 1/2$ and $\Delta t = 1/4$ case compared to the other estimated cost rates g_D^{den}, g_D^{int}, and g_D^{uni} which converge similarly to the $\Delta t = 1$ case depicted in Figure 1.

In any real-life application of MDP for maintenance optimization there is inevitably going to be a model inaccuracy in that q_i will never perfectly capture the true deterioration dynamics of the components in the system. However, the fact that we obtain a near-optimal policy even with inaccurate transition probabilities is reassuring. Furthermore, the results in Figure 1 suggest that dynamic programming for optimization of systems with several components is feasible, since we only need few discrete states for each component. From the perspective of the practical use of MDP, our results also indicate that estimating discrete transition probabilities from data via Equation (8) is a sound approach, because the cost rate estimate we get from solving the corresponding MDP is accurate.

5.2. Multigrid algorithm

When the objective is to approximate the optimal policy, π^*, in the continuous-state MDP formulation of the problem, we can exploit the fact that the q_i^{exp} method approximates the continuous deterioration process better than the other methods. The one-way multigrid algorithm analyzed in [7] is a method for approximating π^* by solving multiple discrete MDP with an increasingly fine discretization (grid). When an MDP with D intervals has been solved, the resulting value function, v_D, is used as the initialization of the algorithm for solving the MDP with $2D$ discretization

	without multigrid		multigrid	
method	runtime	#iterations	runtime	#iterations
q_i^{den}	10.29sec	18	8.71sec	13
q_i^{int}	10.22sec	18	8.37sec	12
q_i^{uni}	10.39sec	18	7.70sec	12
q_i^{exp}	10.22sec	18	6.45sec	10

Table 1: Runtime and number of iterations for different discretization methods when using $D = 64$, VI with the average reward optimality criterion and $\epsilon = 0.001$.

intervals. This leads to a shorter total computational time, since most of the work is being carried out on the coarse discretization MDPs. When using the q_i^{exp} probabilities, the value function, v_D, is very close to its asymptotic value already at $D = 4$, so it is plausible that the subsequent steps in the multigrid algorithm will be faster using this method compared to using q_i^{den}, q_i^{int}, or q_i^{uni}.

Table 1 shows the runtime and number of iterations of solving the example problem using the multigrid algorithm starting with $D = 2$ and ending at $D = 64$. At each D, the VI algorithm with the average reward optimality criterion, and $\epsilon = 0.001$ is used. The multigrid runtimes are the accummulated time required for solving all MDPs with $D \in \{2, 4, 8, 16, 32, 64\}$, while the number of iterations are only for the $D = 64$ MDP since these account for most of the total runtime. As Table 1 shows we can solve the $D = 64$ problem somewhat faster when using q_i^{exp}. The relative benefit is, however, greater when solving MDPs under the expected total discounted reward optimality criterion. This is shown in Table 2. Compared to the convergence criterion in Equation (12) for the average reward case, the convergence criterion in Equation (15) is less sensitive, which is the reason for the runtimes in Table 2 being a lot higher than in Table 1. The Gauss-Seidel version of VI is a special case of an asynchronous dynamic programming algorithm, which is better suited for parallelization [18] and can sometimes be faster than using the standard VI with span norm stopping criterion in Equation (12) [14]. For $\epsilon = 0.001$ the relative decrease in runtime when using q_i^{exp} is roughly the same as for the average reward case. However, using a tolerance of $\epsilon = 1$, the runtime is suddenly an order of magnitude lower. This suggests that when solving the final $D = 64$ MDP, which is initialized with the $D = 32$ solution, each method uses the same number of iterations when taking the value function from an accuracy of 1 to 0.001 and that the effect of

	$\epsilon = 0.001$		$\epsilon = 1$	
method	runtime	#iterations	runtime	#iterations
q_i^{den}	525.29sec	798	137.65sec	193
q_i^{int}	515.36sec	760	114.54sec	157
q_i^{uni}	511.49sec	761	117.42sec	158
q_i^{exp}	356.53sec	527	12.60sec	16

Table 2: Runtime and number of iterations for the multigrid algorithm for different discretization methods when using $D = 64$, Gauss-Seidel VI and the expected total discounted reward optimality criterion.

a more accurate initialization is gone at this point. For $\epsilon = 1$ and $\epsilon = 0.001$ the policies π_D^{exp} are identical up to $D = 16$ and almost identical for $D = 32$ and $D = 64$. Considering this, and that we will still have a slight discretization error at $D = 64$, it is not worth solving the MDP with a low tolerance, ϵ, as the computational requirements are a lot higher.

6. Conclusion

In this paper we compare different methods of discretization, which is necessary when using dynamic programming for optimizing maintenance of continuously deteriorating components. Even though the transition probabilities obtained from each method are different, the policies we obtain in the considered example are already nearly identical when the discretization is very coarse. Additionally, the number of discretization intervals needed to obtain a near-optimal policy is also quite low for all methods. Due to the discretization error, the value of the policy we find with dynamic programming is only an estimate of its value w.r.t. the continuous deterioration process. As our results indicate, this estimate can be inaccurate depending on how transition probabilities between states are defined. We use Monte Carlo simulation of the continuously deteriorating system to reveal how large the error is. A scheme based on the expected number of transitions between discretization intervals, is the only method among the ones we tested that accurately estimates the performance of the policy w.r.t the continuous deterioration process. We show how this scheme can be evaluated exactly rather than via Monte Carlo estimation.

Condition monitoring is sometimes done with a categorization into a few qualitatively labeled states such as “new”, “worn”, and “failed”. The scheme based on the expected number of trans-

itions between discretization intervals is conceptually the same as estimating transition probabilites between such qualitative states directly from field data. One may fear that a very simple deterioration model might hamper the performance of the policy obtained from the optimization algorithm. The fact that only a few intervals are needed to obtain near-optimal performance in our example is therefore an indication that this is not necessarily the case.

The accurate discretization scheme allowed us to solve the MDP faster when using a multigrid algorithm. In this procedure, the MDP is solved repeatedly, using the solution from coarse discretizations to initialize the algorithm on finer discretizations. Using the accurate discretization method, the value function is close to its asymptotic value already at coarse discretization levels. Therefore, fewer iterations are needed before convergence, when the MDP is solved with finer levels of discretization.

Throughout the study, we use a piece-wise flat extension of the policies obtained from the discretized problem to the state space of the continuous deterioration process. This was done to mimic a practical application, where the condition is not measured exactly but based on a coarse categorization of condition levels. Assuming the true continuous deterioration is observed, there are more advanced techniques for extending the discrete-state optimal policy to the state space of the continuous deterioration process [16]. An interesting continuation of our study is to investigate how much these techniques improve the solution for the problem we considered.

References

[1] S. Alaswad, Y. Xiang, A review on condition-based maintenance optimization models for stochastically deteriorating system, Reliability Engineering and System Safety 157 (2017) 54–63. doi:10.1016/j.ress.2016.08.009.

[2] K. Golabi, R. B. Kulkarni, G. B. Way, A Statewide Pavement Management System, INFORMS Journal on Applied Analytics 12 (6) (1982) 5–21. doi:10.1287/inte.12.6.5.

[3] E. Byon, Y. Ding, Season-dependent condition-based maintenance for a wind turbine using a partially observed markov decision process, IEEE Transactions on Power Systems 25 (4) (2010) 1823–1834. doi:10.1109/TPWRS.2010.2043269.

[4] T. M. Welte, J. Vatn, J. Heggset, Markov State Model for Optimization of Maintenance and Renewal of Hydro Power Components, in: 2006 International Conference on Probabilistic Methods Applied to Power Systems, IEEE, 2006, pp. 1–7. doi:10.1109/PMAPS.2006.360311.

[5] R. Arismendi, A. Barros, A. Grall, Piecewise deterministic Markov process for condition-based maintenance models — Application to critical infrastructures with discrete-state deterioration, Reliability Engineering and System Safety 212 (2021) 107540. doi:10.1016/j.ress.2021.107540.

[6] J. M. van Noortwijk, A survey of the application of gamma processes in maintenance, Reliability Engineering and System Safety 94 (1) (2009) 2–21. doi:10.1016/j.ress.2007.03.019.

[7] C.-S. Chow, J. N. Tsitsiklis, An optimal one-way multigrid algorithm for discrete-time stochastic control, IEEE transactions on automatic control 36 (8) (1991) 898–914. doi:10.1109/9.133184.

[8] J. Rust, Using Randomization to Break the Curse of Dimensionality, Econometrica 65 (3) (1997) 487–516. doi:10.2307/2171751.

[9] S. Özekici, Optimal periodic replacement of multicomponent reliability systems, Operations Research 36 (4) (1988) 542–552. doi:10.1287/opre.36.4.542.

[10] A. H. Elwany, N. Z. Gebraeel, L. M. Maillart, Structured Replacement Policies for Components with Complex Degradation Processes and Dedicated Sensors, Operations research 59 (3) (2011) 684–695. doi:10.1287/opre.1110.0912.

[11] C. P. Andriotis, K. G. Papakonstantinou, Managing engineering systems with large state and action spaces through deep reinforcement learning, Reliability Engineering & System Safety 191 (2019) 106483. doi:10.1016/j.ress.2019.04.036.

[12] J. Xu, Z. Liang, Y.-F. Li, K. Wang, Generalized condition-based maintenance optimization for multi-component systems considering stochastic dependency and imperfect maintenance, Reliability Engineering & System Safety 211 (2021) 107592. doi:10.1016/j.ress.2021.107592.

[13] Q. Sun, Z.-S. Ye, N. Chen, Optimal Inspection and Replacement Policies for Multi-Unit Systems Subject to Degradation, IEEE Transactions on Reliability 67 (1) (2018) 401–413. doi:10.1109/TR.2017.2778283.

[14] J. F. Andersen, A. R. Andersen, M. Kulahci, B. F. Nielsen, A numerical study of Markov decision process algorithms for multi-component replacement problems, European Journal of Operational Researchdoi:10.1016/J.EJOR.2021.07.007.

[15] K. G. Papakonstantinou, M. Shinozuka, Planning structural inspection and maintenance policies via dynamic programming and Markov processes. Part II: POMDP implementation, Reliability Engineering & System Safety 130 (2014) 214–224. doi:https://doi.org/10.1016/j.ress.2014.04.006.

[16] J. P. Rust, A Comparison of Policy Iteration Methods for Solving Continuous-State, Infinite-Horizon Markovian Decision Problems Using Random, Quasi-Random, and Deterministic Discretizations (1997). doi:https://dx.doi.org/10.2139/ssrn.37768.

[17] B. de Jonge, Discretizing continuous-time continuous-state deterioration processes, with an application to condition-based maintenance optimization, Reliability Engineering and System Safety 188 (March) (2019) 1–5. doi:10.1016/j.ress.2019.03.006.

[18] D. P. Bertsekas, Distributed asynchronous computation of fixed points, Mathematical Programming 27 (1) (1983) 107–120. doi:10.1007/BF02591967.

CHAPTER 5

TBM versus CBM

In the previous two chapters we have only focused on the computational aspects and modeling aspects of the optimization. Little attention has been given to the actual policies we obtain, their structure, or their performance in terms of costs. In chapter 1, we mentioned that one purpose of maintenance studies, is to provide general guidelines for good maintenance practices for various systems and circumstances. The following paper falls into this category of maintenance studies. Specifically, the purpose is to investigate the performance of optimal TBM and CBM policies in a multi-component system, and how the difference in performance changes with the configuration of the system parameters.

The gaining interest in CBM from both practitioners and researches is founded in a rationale that CBM generally outperform traditional TBM. While this is true in many situations, there are also cases where there is little or no benefit (Jonge et al. 2017; Shin and Jun 2015). Given the practical difficulty of establishing a condition monitoring process and modeling the deterioration, it is therefore relevant to study, which kinds of systems benefit the most from a CBM policy. Indeed, there is also a high cost associated with implementing CBM (Shin and Jun 2015), so the decision to do so should of course be done with consideration to the possible long term benefits.

There have been other studies that compare CBM and TBM, which we describe in more detail in the paper, but most of these studies focus on single-component systems. The few studies that do consider multi-component systems only perform a superficial comparison, and they do not attempt to derive any general insights. The sensitivity analysis we conduct is a contribution in this direction.

We make the comparison between CBM and TBM based on the unifying modeling framework from Chapter 3, however, we consider a special case of the system we presented there. In this chapter, we consider a simpler cost structure of the system, where we omit the system failure cost, c_f, and the structural dependence from a K-out-of-N reliability structure. Instead, each component has a preventive and corrective replacement cost, and must be replaced upon failure. For the K-out-of-N structure we assumed previously, it was sometimes optimal to let a component fail and then never replace it again. The same behavior is observed in Olde Keizer et al. (2018), where a K-out-of-N system is also considered. This behavior would make a comparison between CBM and TBM more intricate, which is why we limit the scope of the analysis to a simpler multi-component system. Without the structural dependence, a more appropriate name would be a multi-asset system. We can imagine the system in

this chapter being an offshore wind turbine farm, where each turbine may fail without affecting the state of the others. In this system, there is still stochastic dependence stemming from the same operating environment, and economic dependence via a joint setup cost, which represent the travel of the maintenance crew to the offshore location.

The reader may notice that the largest system we consider in this chapter has four components, while the largest number of components in Chapter 3 was five and seven for CBM and TBM, respectively. The reason for this is that the truncation of the state space in the TBM MDP model is determined by the how long a component will be functioning in terms of number of decision epochs. In Chapter 3, the parameters of the gamma process were chosen such that at most 17 epochs would pass before a component would have failed with high probability. In this chapter, we choose these parameters such that we need between 58 and 114 epochs, hence the TBM MDP state spaces are much larger in this chapter. The reason for the difference choice is that we want to ensure a higher level of flexibility in the set of admissible TBM policies, which we also discuss in detail in the following manuscript. Here, we simply note the fact that at most four components are solved in this chapter does not compromise the results or conclusions from Chapter 3.

5.1 Paper C

A comparative study of time-based maintenance and condition-based maintenance for multi-component systems*

Jesper Fink Andersen[a,*], Murat Kulahci[a,b], Bo Friis Nielsen[a]

[a] *Technical University of Denmark, Department of Applied Mathematics and Computer Science, Anker Engelunds Vej 1, 2800 Kgs. Lyngby, Denmark*
[b] *Luleå University of Technology, Department of Business Administration, Technology and Social Sciences, Luleå, Sweden*

Abstract

In this paper, we investigate the benefit of Condition-Based Maintenance (CBM) over Time-Based Maintenance (TBM) in a multi-component system. We consider a system, where the components deteriorate according to a gamma process, with stochastic dependence between components modeled with a Clayton-Lévy copula function. Economic dependence between components is modeled as a joint setup cost for replacing components. Optimal CBM and TBM policies are identified by formulating the two cases as separate Markov decision processes and using dynamic programming for optimization. We compare the optimal TBM and CBM policies under varying system parameter configurations, using the average cost per time unit as a measure of performance. The results from our numerical experiments show that the performance difference between the TBM policy and the CBM policy decreases as the setup cost increases, and the difference increases when the degree of stochastic dependence increases. Furthermore, when the number of components increase, the difference between TBM and CBM become less sensitive to the ratio between the preventive replacement cost and the corrective replacement cost. Overall, the benefit of CBM over TBM is smaller in the multi-component system compared to the single-component system.

Keywords: Time-based maintenance, Condition-based maintenance, Multi-component system, Markov decision process

* *Declarations of interest*: none

* Corresponding author
Email addresses: jfan@dtu.dk (Jesper Fink Andersen), muku@dtu.dk (Murat Kulahci), bfni@dtu.dk (Bo Friis Nielsen)

1. Introduction

For engineering systems that are subject to deterioration and failures, two major categories of maintenance strategies exist; Time-Based Maintenance (TBM) and Condition-Based Maintenance (CBM). As the names suggest, the decision to maintain the system is based on the elapsed time in TBM and the physical condition of the system in CBM.

Mathematical optimization models for both concepts date back to the early 1960's (Derman, 1963; Barlow & Proschan, 1996). A prerequisite for performing CBM is a mechanism for monitoring the deterioration of the system. The practical application of CBM was held back for a long time by a lack of methods for condition monitoring (Dekker, 1996), but there has been a steadily growing interest in CBM in industry over the last couple of decades. This is partly due to the increasing need for better maintenance strategies for the more expensive and more complex systems that exist today. Secondly, recent developments in sensor technology and condition monitoring techniques have also made CBM more accessible to a larger segment (Alaswad & Xiang, 2017; Shin & Jun, 2015).

TBM and CBM policies exist in varying degrees of sophistication. The two simplest forms of TBM are the Block Replacement (BR) policy and the Age Replacement (AR) policy. Under the BR policy, periodic maintenance is scheduled according to calendar time, and under the AR policy, maintenance is performed whenever the system reaches an age threshold or fails (Barlow & Proschan, 1996). A simple CBM policy is a Periodic Inspection and Replacement (PIR) policy that prescribes maintenance actions, whenever the observed deterioration of the system exceeds a predetermined threshold. It is common to describe the deterioration of the system with a stochastic process, and then model inspections by assuming the points in time where this process is observed. In this study, we use the gamma process to model deterioration, a commonly used process in studies of maintenance policies (van Noortwijk, 2009).

Both CBM and TBM attempt to balance the cost of frequent maintenance activities with the cost of system failures. When the deterioration of the system is gradual, CBM has an advantage over TBM, since maintenance can be performed just before a failure occurs in the former. If a cost is associated with inspecting the condition of the system, CBM may no longer be the cheapest strategy, since TBM does not involve condition monitoring. Furthermore, the benefit of CBM over TBM diminishes if the deterioration process is close to one of the two extremes of either being

completely deterministic or completely unpredictable. The latter implies that failure is induced by a single jump in the process occurring after an exponentially distributed time, in which case neither TBM nor CBM can prevent the failure (de Jonge et al., 2017). It is therefore not always obvious that maintenance managers should pursue an implementation of CBM, because the long-term cost savings may not be very large. From a practical point of view, TBM is simpler to implement as CBM has an investment cost for setting up the condition monitoring process (Shin & Jun, 2015).

Previous studies have considered the relative performance of TBM and CBM when costs and deterioration process parameters are varied. From the review given in de Jonge et al. (2017), the focus has almost exclusively been on single-component systems. In de Jonge et al. (2017), the PIR policy and the AR policy described above are compared for a single-component system. The authors investigate the practical factors of planning time, imperfect condition information, and an uncertain deterioration failure limit. Cherkaoui et al. (2018) also consider a single component, and they use the BR policy as the representative for TBM. They consider two CBM policies, a periodic inspection PIR policy, and a variation of the PIR policy where the next inspection is chosen such that a given level of reliability of the system is ensured. The main focus in Cherkaoui et al. (2018), is the robustness of policies, both in terms of the cost variability and also the expected cost when the policy is at a sub-optimal configuration. In uit het Broek et al. (2021), BR and PIR are compared for a single-component production system, when the policies are jointly optimized with a controllable production rate.

Regarding multi-component systems, we have only found two studies that compare TBM and CBM. In Bouvard et al. (2011), a rolling-horizon optimization of a heuristic policy is developed, and the authors compare the case were condition information is utilized to the case where only age information is utilized. They consider a 3-component numerical example with fixed cost parameters, and vary the inspection interval and deterioration parameters. As such, no attempts are made to investigate the aspects that are particularly relevant in multi-component systems, namely stochastic, structural, and economic dependence between components.

The AR, BR, and PIR policies have all been studied extensively and in many variations Wang (2002). Therefore, they are the natural choice as representative TBM and CBM policies in a TBM and CBM comparison for a single-component system. All three policies are examples of *policy optimization* (Nicolai & Dekker, 2008), where the solution is searched for within a parameterized

class of policies. This is also a common approach for optimization in multi-component systems. The second multi-component TBM and CBM comparison is found in Olde Keizer et al. (2016). The authors consider a K-out-of-N system with economic dependence through a joint maintenance setup cost. The authors compare an optimal CBM policy found via dynamic programming to six different parameterized policies. The six reference policies are: 1) replace components only upon failure, 2) individual AR for each component, 3) BR of all components simultaneously, 4) BR with intermediate replacements of failed components, 5) individual PIR for each components, and 6) an (m_i, M_i) multi-threshold CBM policy, where component i is replaced either correctively upon failure, preventively when the deterioration exceeds M_i, or opportunistically if deterioration exceeds m_i and another component is also being replaced. Policies 2)-4) are TBM and policies 5)-6) are CBM. Even though the sensitivity analysis includes varying the setup cost and the number of components, the main focus of the comparison is to demonstrate the benefit of using a policy that exploit the K-out-of-N reliability structure over the six heuristics that are easier to implement in practice. The comparison mainly revolves around the optimal CBM policy compared to policy 6), because the latter is consistently the best out of the six reference policies.

None of the TBM policies 2)-4) in Olde Keizer et al. (2016) exploit the possibility of clustering maintenance. The (m_i, M_i) policy does so through the threshold m_i, and this policy can also be an extension of AR, when m_i and M_i represent age thresholds. A more relevant comparison of TBM and CBM would therefore be between the PIR version and the AR version of the (m_i, M_i) policy. There are, however, many possible extensions of the basic AR, BR and PIR policies besides the (m_i, M_i) policy (Dekker et al., 1997). In multi-component systems it is generally the case that such extensions do not contain the globally optimal policy (Nicolai & Dekker, 2008). We believe a fair comparison between TBM and CBM requires that the chosen representative policies are then "equally suboptimal" compared to their respective globally optimal policies. It is impossible to check this without also computing the globally optimal policies, which is why we focus on comparing the latter in this study.

When we say globally optimal policy, we mean optimal within a space of policies with as few restrictions as possible, given the available information and a set of assumptions about the system. We obtain such policies, by formulating both the TBM and the CBM problem as Markov Decision Processes (MDPs) and solving them with Dynamic Programming (DP). We assume a

continuous deterioriation process, i.e., the gamma process, such that the variance of the component deterioration and the degree of stochastic dependence between components can be adjusted easily in the sensitivity analysis. General purpose DP algorithms can only handle discrete-time and discrete-space MDPs, but through sufficiently fine discretization of both time and space, we obtain policies that closely approximate the globally optimal policies.

We consider a multi-component system that has economic dependence through a joint maintenance setup cost and stochastic dependence modeled via a Lévy copula function. We investigate the effects of varying aspects that pertain to multi-components systems, namely, the number of components, the setup cost, and the degree of stochastic dependence between components. For each system configuration, we compare the optimal TBM and CBM policies via the long-run average maintenance cost per time unit, which we also refer to as the cost rate. It is currently unknown if and how the above-mentioned aspects affect the relative performance of TBM and CBM. As the relevance of multi-component models is growing with the increasing complexity of modern industrial systems, this study is a contribution that provides relevant insights to both maintenance practitioners and theoreticians alike.

The remainder of this paper is structured as follows. In Section 2 we define the multi-component system and formulate the problem of replacing components as a CBM MDP model and as a TBM MDP model. In Section 3 we briefly review the DP algorithm used to obtain optimal policies in the MDPs. In Section 4 we present and discuss the result from the sensitivity analysis, and in Section 5 we provide a conclusion to the study.

2. Model description

We consider a system with N components each of which are subject to deterioration. In order to keep the system in operation, the components can be replaced with new and identical ones. We let $\mathbf{X}_t = (X_t^1, \ldots, X_t^N)$, $t \geq 0$, be a stochastic process that describes the condition of the deteriorating system, when no component replacements are performed. The condition of the maintained system, here denoted $\mathbf{Y}_t = (Y_t^1, \ldots, Y_t^N)$ is the process, with the same increments $\mathbf{Y}_{t_2} - \mathbf{Y}_{t_1} = \mathbf{X}_{t_2} - \mathbf{X}_{t_1}$, $0 \leq t_1 < t_2 < \infty$, when no replacements are done in the interval $[t_1, t_2]$, and for which a replacement, say of component i at time t, leads to $Y_t^i = 0$.

We make the following assumptions about the system:

A1 For all $i \in \{1, \ldots, N\}$, the marginal deterioration process $\{X_t^i\}_{t \geq 0}$, is a time-homogeneous gamma process. This process is a non-decreasing Lévy process for which $X_0^i = 0$, and where increments $X_t^i - X_s^i$, $s < t$, are gamma distributed with shape parameter $\alpha(t-s)$ and rate parameter β. Hence, the deterioration processes for the individual components have identical probability laws.

A2 We assume that decisions to replace components are made at discrete points in time, $\tau\Delta$, $\tau = 0, 1, \ldots$. We refer to the time $\tau\Delta$ as decision epoch τ, and we assume the time between epochs, $\Delta > 0$, is fixed.

A3 Component i is assumed to fail when $Y_t^i \geq L$, where L is a fixed failure level. Failures are self-announcing in the sense that they are known at the following decision epoch.

A4 All replacements are perfect and instantaneous. Upon a replacement of component i at epoch τ we have $Y_{\tau\Delta}^i = 0$.

A5 A component can be replaced at any epoch, τ. If the component is replaced preventively, i.e., when $Y_{\tau\Delta^-}^i < L$, we the cost of replacement is c_r. If the component has failed, $Y_{\tau\Delta^-}^i \geq L$, it must be replaced correctively at a cost $c_r + c_b$, where c_b is an additional breakdown cost.

A6 Whenever at least one component is replaced we incur a joint setup cost, c_s.

A7 In the TBM case, we assume that the age of each component is known at each epoch, that is, the number of epochs since the last replacement.

A8 In the CBM case, we assume that the (discretized) deterioration level is known at each epoch.

We formulate the TBM version and the CBM version of the replacement problem as MDPs, and refer to these as the TBMDP and CBMDP model, respectively. The TBMDP and CBMDP models we construct are multi-component extensions of the base model in de Jonge et al. (2017), where all the same system assumptions are made (except **A6** and the discretization part of **A8**). There are some subtle yet important implications of the assumptions **A1**-**A8** on the scope of the comparison between TBM and CBM, some of which are not addressed in de Jonge et al. (2017). It is therefore appropriate to first address some conceptual differences of the TBM and CBM problem under these and alternative assumptions.

For the TBM problem, the discrete epochs in assumption **A2** imply a discrete state space in the TBMDP, since this state space contains the possible ages of components. For the CBM problem we also construct a discrete-state MDP, by discretization of the state spaces of the gamma deterioration processes, $\mathbf{X}_t$. Because we consider a multi-component system, the discrete-time and discrete-state MDP formulation is required for optimization with DP algorithms. While Haurie & L'Ecuyer (1982) consider a continuous-time discrete-state formulation and Özekici (1988) considers a discrete-time continuous-state formulation of multi-component replacement problems, these models are analyzed in order to characterize the structure of the optimal policies. In both studies, the authors conclude that during computation one must approximate the optimal policy with a solution from a discrete model, and this is also the approach we use here.

Assumption **A2** is somewhat restrictive for the TBM case. For instance, when $N = 1$ the optimal AR policy is characterized by a threshold value for preventive replacement (Barlow & Proschan, 1996), and this may very well not be a multiple of Δ, but can be any value in the interval $(0, \infty)$. However, by setting Δ very low compared to the expected life of the components, the discrete-time formulation may be regarded as a good approximation to the continuous-time problem.

In a CBM setting, assumptions **A2** and **A8** correspond to periodic inspection of the component condition. This discrete-time formulation is more natural in the CBM case, since continuous inspection of a gradual deterioration process allows the decision-maker to always replace components just before failure occurs. This is hardly an interesting problem, and therefore CBM studies with continuous inspections assume additional complications such as L being a random variable (Zuckerman, 1978), a condition-dependent failure rate (Park, 1988b), or a system with both monitored and non-monitored components (Castro et al., 2020).

Since the deterioration process is continuous in time, a component failure may occur at some time t_{fail}, between two epochs, say $(\tau - 1)\Delta$ and $\tau\Delta$. By assumption **A3** the component is known to have failed at epoch τ in both the TBM case and the CBM case. We call this self-announcing failures, albeit it is with a delay of $\tau\Delta - t_{fail}$ due to the time discretization.

In a CBM context, the $\tau\Delta - t_{fail}$ delay usually follows from an assumption that failure are not self-announcing and that they are only detected upon inspection. It is then common to include a downtime cost proportional to $\tau\Delta - t_{fail}$ (Xu et al., 2021; Sun et al., 2018; Grall et al., 2002;

Abdel-Hameed, 1987). The downtime cost often appears together with a cost of inspecting the system condition. The inspection interval, Δ, is then considered a decision variable that must be optimized. In Cherkaoui et al. (2018), the PIR policy is jointly optimized with the inspection interval for a single-component system and is then compared to the BR policy.

The BR policy is a suitable TBM representative policy when failures are not self-announcing. The BR policy is static, in the sense that all replacement times are prescheduled. The AR policy, on the other hand, is dynamic because the replacement times depend on when failures of components occur. We therefore see the case of self-announcing failures and TBM policies based on component ages as the more interesting case to compare with CBM.

Park (1988a) considers a single-component CBM system with periodic inspections and self-announcing failures. When the component fails, it is replaced immediately and a new inspection cycles start. In the present study, we regard the CBMDP model as a multi-component extension of the model proposed by Park (1988a). However, just as for the TBMDP model, the replacement delay $\tau\Delta - t_{fail}$ appears as a time discretization error. For a fixed value of Δ this error is of equal magnitude in the CBM and the TBM case, hence the comparison is still meaningful.

2.1. MDP formulation

At each epoch, the following events take place in an MDP. The system occupies a state $\mathbf{s} \in S$, where S is a finite set; an action $\mathbf{a} \in A_{\mathbf{s}}$ is chosen, where $A_{\mathbf{s}}$ is also a finite set; a reward, $r(\mathbf{s}, \mathbf{a})$ is received; and the system transitions to a new state, $\mathbf{s}' \in S$, with probability $p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$.

In both the CBMDP and TBMDP, the state is a vector with N elements, $\mathbf{s} = (s_1, \ldots, s_N)$, where $s_i \in \{0, \ldots, D\}$, and the state space is given by $S = \{0, \ldots, D\}^N$. Component i is new when $s_i = 0$, and $s_i = D$ means it has failed. The interpretation of s_i and D for the CBMDP and TBMDP are given in Sections 2.1.1 and 2.1.2, respectively.

An action is a binary vector $\mathbf{a} = (a_1, \ldots, a_N)$, where a_i equal to 1 (0) indicates component i is replaced (not replaced). By assumption **A5**, a failed component must be replaced, so the set of possible actions in occupied state, $\mathbf{s} \in S$, is given by $A_{\mathbf{s}} = \{\mathbf{a} \in \{0, 1\}^N : a_i = 1 \text{ if } s_i = D\}$.

In both MDP models, the reward function is given by

$$r(\mathbf{s}, \mathbf{a}) = \sum_{i=1}^{N} a_i \left(c_r + c_b \mathbf{1}_{s_i = D}\right) + c_s \left(1 - \prod_{i=1}^{N} (1 - a_i)\right), \tag{1}$$

where $\mathbf{1}_{s_i=D}$ is the indicator function for the event $s_i = D$. The first term in Equation (1) account for the replacement cost and breakdown cost introduced in assumption **A5**, and the second term is the setup cost from assumption **A6**.

By assumption **A4**, at epoch $\tau\Delta$ an action, $\mathbf{a} \in A_{\mathbf{s}}$, instantly moves the system from state $\mathbf{s}$ to a *post-decision* state, $((1-a_1)s_1, \ldots, (1-a_N)s_N) \in S$, where the elements corresponding to replaced components now have the value zero. From the post-decision state and still at epoch $\tau\Delta$, the system then transitions to the state $\mathbf{s}' \in S$ at epoch $(\tau+1)\Delta$ with probability $p(\mathbf{s}'|((1-a_1)s_1, \ldots, (1-a_N)s_N), \mathbf{0})$, where $\mathbf{0} \in A_{\mathbf{s}}$ is the action of not replacing any components. Instead of specifying $p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$ for all $\mathbf{s}, \mathbf{s}' \in S$ and $\mathbf{a} \in A_{\mathbf{s}}$, it is therefore enough to specify the probabilities $q(\mathbf{s}'|\mathbf{s}) = p(\mathbf{s}'|\mathbf{s}, \mathbf{0})$.

2.1.1. CBMDP

In this model, the state vector, $\mathbf{s} \in S$, represents a discretized version of the condition of the maintained system, $\mathbf{Y}_t$. For the discretization, we divide the interval $[0, L)$ into D equally sized intervals $I_k = [kL/D, (k+1)L/D)$, $k = 0, \ldots, D-1$, and let $I_D = [L, \infty)$. An element $s_i = k$, $k \in \{0, \ldots, D\}$, of the state vector, $\mathbf{s} \in S$, then corresponds to $Y_t^i \in I_k$. The CBMDP occupying state $\mathbf{s} \in S$ at epoch τ therefore corresponds to $\mathbf{Y}_{\tau\Delta} \in \mathbf{I}_{\mathbf{s}}$, where $\mathbf{I}_{\mathbf{s}} = I_{s_1} \times \cdots \times I_{s_N}$.

We define the transition probabilities between the discrete states $\mathbf{s}, \mathbf{s}' \in S$ by

$$q(\mathbf{s}'|\mathbf{s}) = P(\mathbf{X}_{(\tau+1)\Delta} \in \mathbf{I}_{\mathbf{s}'} | X_{\tau\Delta}^i = (s_i + 1/2)L/D, i = 1, \ldots, N). \tag{2}$$

This is the probability that the deterioration process, $\mathbf{X}_t$, advances to the region $\mathbf{I}_{\mathbf{s}'}$ in epoch $\tau+1$ given each component had a deterioration level equal to the midpoint of their respective origin interval, I_{s_i}, at epoch τ. In the case of independent component deterioration, the probability in Equation (2) can be calculated as the product

$$q(\mathbf{s}'|\mathbf{s}) = \prod_{i=1}^{N} P(X_{(\tau+1)\Delta}^i \in I_{s_i'} | X_{\tau\Delta}^i = (s_i + 1/2)L/D), \tag{3}$$

where each factor in Equation (3) can be evaluated from the gamma distribution function of the increment $X_{(\tau+1)\Delta}^i - X_{(\tau)\Delta}^i$. The case of dependent component deterioration is discussed in Section 2.1.3.

2.1.2. TBMDP

In this section, we construct an MDP for the situation described in assumption **A7**, where the condition of the maintained system, $\mathbf{Y}_t$, is not observed, with the exception that failures are self-announcing in accordance with assumption **A3**. In this model, the element, $s_i \in \{0, \ldots, D-1\}$, of the state vector, $\mathbf{s} \in S$, represents the age of component i measured as the number of epochs since the last replacement. The value $s_i = D$ indicates that the component has failed. In principle, it is possible for a component to reach an age higher than D without failing. However, general DP algorithms, like the one we describe in Section 3, only work for finite state spaces. Therefore, D is also a truncation point for the maximum age a component can attain in the TBMDP model. As long as the value of D is chosen large enough, the optimal TBM policy can be identified from the TBMDP. If, however, D is set to a low value, it may be better to let components age for longer than D epochs. In this case, the policy we obtain from solving the TBMDP cannot be optimal, as the set of admissible policies has been restricted too much by the truncation. We avoid this by choosing D such that the probability of a component surviving for D epochs, $P(X^i_{D\Delta} < L)$, is close to zero, which we describe further in Section 4.

For the case of independent component deterioration, the transition probabilities in the TBMDP are defined as follows. At each transition, a component in state $s_i < D$ can either fail and transition to state D, or age by one time unit, thereby transitioning to state $s_i + 1$. Let $q_i(s'|s)$, $s, s' \in \{0, \ldots, D\}$, denote the probability of component i transitioning from s to s'. We define this probability as

$$q_i(s'|s) = \begin{cases} P(X^i_{(s+1)\Delta} \geq L | X^i_{s\Delta} < L) & s' = D, s < D-1 \\ P(X^i_{(s+1)\Delta} < L | X^i_{s\Delta} < L) & s' = s+1 < D \\ 1 & D-1 \leq s \leq s' = D \\ 0 & \text{else.} \end{cases} \tag{4}$$

The first line in Equation (4) is the probability that a functioning component of age s fails, and the second line is the probability that it does not fail. When a component has age $s = D-1$ or is failed, $s = D$, it will be in state D at the next epoch with certainty, which is the content of the third line. We can now define the transition probability from the vector state $\mathbf{s} \in S$ to state $\mathbf{s}' \in S$ as

$$q(\mathbf{s}'|\mathbf{s}) = \prod_{i=1}^{N} q_i(s'_i|s_i). \tag{5}$$

2.1.3. Stochastic dependence

In Sections 2.1.1 and 2.1.2 we formulated the CBMDP and TBMDP models for the case, where the marginal deterioration processes, $X_t^1, \ldots, X_t^N$, are independent. In Section 4, we also investigate the effects of dependence between the marginal deterioration processes, which in a broad term is known as stochastic dependence (de Jonge & Scarf, 2020). We model this dependence through a Clayton-Lévy copula function, which has previously been used in the context of deterioration modeling (Shi et al., 2020; Li et al., 2016; Jiang et al., 2021). When this dependence is introduced, Equations (3) and (5) for the MDP transition probabilities are no longer valid. Instead we use the Monte Carlo estimation procedure described in Andersen et al. (2021). For the sake of brevity, we will not repeat the full description of the procedure here, but simply note that the Clayton-Lévy copula function contains a parameter, $\theta > 0$, and the larger the value of θ, the bigger is the probability that large jumps in processes $X_t^1, \ldots, X_t^N$ occur simultaneously. The MDP transition probability estimation procedures described in Andersen et al. (2021) are based on the simulation algorithm for Lévy processes with copula dependence developed in Grothe & Hofert (2015).

3. Dynamic programming

The optimal policies in the MDP models are found using the Value Iteration (VI) algorithm. In the set of all mappings from S to A, Π, the VI algorithm identifies a policy, $\pi \in \Pi$, that maximizes the average reward per time unit, i.e. the cost rate, by successively computing for $n \in \mathbb{N}_0$

$$v_{n+1}(\mathbf{s}) = \max_{\mathbf{a} \in A} \left\{ r(\mathbf{s}, \mathbf{a}) + \sum_{\mathbf{s}' \in S} p(\mathbf{s}'|\mathbf{s}, \mathbf{a}) v_n(\mathbf{s}') \right\} \quad \forall \mathbf{s} \in S, \tag{6}$$

where $v_n : S \to \mathbb{R}$ is the value function. The process is stopped when

$$\max_{\mathbf{s} \in S} \{v_{n+1}(\mathbf{s}) - v_n(\mathbf{s})\} - \min_{\mathbf{s} \in S} \{v_{n+1}(\mathbf{s}) - v_n(\mathbf{s})\} < \epsilon, \tag{7}$$

where $\epsilon > 0$ is a accuracy tolerance parameter. When this criterion is met, an approximation to the optimal cost rate is given by

$$g = (\max_{\mathbf{s} \in S} \{v_{n+1}(\mathbf{s}) - v_n(\mathbf{s})\} + \min_{\mathbf{s} \in S} \{v_{n+1}(\mathbf{s}) - v_n(\mathbf{s})\})/2, \tag{8}$$

which is at most $\epsilon/2$ from the true value Puterman (2005). The output policy, π, consists of the maximizing actions in the final calculation of Equation (6).

4. Comparative study

In the following, we compare the cost rates g_{CBM} and g_{TBM} obtained from the optimal policies π_{CBM} and π_{TBM} in the CBMDP and TBMDP, respectively. Specifically, we look at the absolute difference in cost rate, $\tilde{g} = g_{\text{CBM}} - g_{\text{TBM}}$, under various parameter configurations of the system. Because we have formulated the problem as one of maximization with negative cost parameters, we always have $g_{\text{TBM}} < 0$ and $g_{\text{CBM}} < 0$. Furthermore, $g_{\text{TBM}} < g_{\text{CBM}}$ and $\tilde{g} > 0$ for all system parameter settings.

All MDPs are solved with the VI algorithm using a tolerance of, $\epsilon = 10^{-5}$. The algorithm is implemented in C++ and all numerical experiments were performed on a Huawei XH620 V3 server node, which has two Intel Xeon Processor 2660v3 with ten 2.60GHz cores each (we only utilize one core) and 128GB RAM.

Because of the discretization of the state-space in the CBM model, the cost rate obtained as output from the VI algorithm is only an approximation. The cost rates g_{CBM} and g_{TBM} are therefore estimated from the average of 10 Monte Carlo simulations of the maintained system process, $\mathbf{Y}_t$, each having a length of 10^8 epochs. In all CBMDPs, the discretization parameter, D, is set to 16. The improvement in g_{CBM} when using finer discretization is very small, and also restricts the number of components, for which we can solve the CBMDP in a reasonable amount of time. The state space truncation, D, in the TBMDP is chosen as the smallest value such that $P(X^i_{D\Delta} < L)$ is less than 10^{-6}, 10^{-3}, and 0.05 for $N \in \{1, 2\}$, $N = 3$, and $N = 4$, respectively. The reason for the dependence on N is that the size of the state space otherwise becomes prohibitively large. The value of D in the TBMDP also depends on the period, Δ, and the gamma process parameters, α and β, and for the values we select below D is in the range 58 to 267.

4.1. Parameter settings

Let g_{CM} denote the cost rate of the purely Corrective Maintenance (CM) policy, where components are replaced only upon failure. All parameter configurations are normalized, such that $g_{\text{CM}} \approx -1$, which allows for an easier interpretation of the individual parameter's effect on the cost rate difference, $\tilde{g}$.

We let two parameters remain the same in all considered system configurations, namely the failure limit, $L = 1$, and the time between epochs $\Delta = 0.02$.

Following Assumption **A1**, we choose the component deterioration parameters, α, and β, such that components are identical. Let $\tau_{\text{fail}} = \min\{k \in \mathbb{N}_0 : X^i_{k\Delta} \geq L\}$, that is, the first epoch after a failure. Then

$$E[\tau_{\text{fail}}] = \sum_{k=1}^{\infty} k\Delta P(\tau_{\text{fail}} \geq k\Delta) = \sum_{k=1}^{\infty} k\Delta P(X^i_{(k-1)\Delta} < L). \tag{9}$$

The values of α are chosen in the range $[0.5, 100]$ and β are chosen such that $E[\tau_{\text{fail}}] = 1$, which is done with a numerical search in Equation (9). By selecting pairs of α and β this way, we get components with varying increment standard deviations, $\sigma = \sqrt{\alpha/\beta^2}$ and with the same average time between replacement under the CM policy.

The components are also assumed to be identical w.r.t. the cost parameters c_r and c_b. We consider all combinations of $c_s \in \{0, -0.1/N, \ldots, -0.5/N\}$, and $c_r, c_b \in \{-0.1/N, \ldots, -0.9/N\}$, such that

$$c_s + c_r + c_b = -\frac{1}{N}. \tag{10}$$

In an N-component system with $E[\tau_{\text{fail}}] = 1$ the cost of a corrective replacement of a single component is given by Equation (10), and therefore $g_{\text{CM}} \approx -1$. The reason g_{CM} is not exactly equal to one, is that multiple components sometimes fail simultaneously, and the setup cost is then only paid once, thus $g_{\text{CM}} \geq -1$. For the chosen values of α, β, and Δ simultaneous failures are rare so g_{CM} is in fact very close to negative one.

There is a trade-off between how small we can set Δ and how large we can set N. Recall from Section 2 that both MDP models have a time-discretization error originating from the combination of continuous deterioration, self-announcing failures, and periodic epochs. We want Δ to be small in order to minimize this error, and also because we do not want to restrict the set of admissible policies in the TBM model too much, by only having epochs at far-spaced component ages. However, a smaller Δ means that the state space of the TBMDP becomes larger. This is because components will reach a higher age in terms of number of epochs before they fail, and therefore the truncation D must be larger. We have chosen $\Delta = 0.02$ and $E[\tau_{\text{fail}}] = 1$, so the average number of epochs between failures is 50. Table 1 shows how Δ affects the cost rate for an $N = 1$ system configuration, when α and β are fixed. By lowering Δ from 0.02 to 0.01 the set of admissible policies is expanded in the TBMDP, however, the cost rate only changes at the fourth decimal place, so the benefit of lowering Δ is negligible. In fact, the cost rate becomes worse as Δ decreases. This is the time discretization error decreasing as the distance between the actual time of failure t_{fail} and the

Δ	g_{TBM}	t^*	g_{CBM}
0.04	-0.64722(0.00008)	0.56	-0.43161(0.00007)
0.02	-0.64808(0.00010)	0.56	-0.4242(0.00007)
0.01	-0.64887(0.00013)	0.55	-0.41976(0.00013)
0.005	-0.64907(0.00020)	0.55	-0.41759(0.00017)

Table 1: Cost rates for varying period lengths, Δ. The value t^* is the age-threshold in the optimal TBMDP policy, π_{TBM}. The system parameters are $N = 1$, $\alpha = 4$, $\beta = 3.46$, $c_r = 0.2$, $c_b = 0.8$. Parentheses indicate standard errors

following epoch τ_{fail} becomes shorter, so the overall time between replacements is shortened as Δ decreases. This is also the case in the CBMDP, but here the cost rate, g_{CBM}, improves because the system condition is inspected more often. As Table 1 shows, the standard errors of the Monte Carlo estimated cost rates, g_{TBM} and g_{CBM}, are very small for the chosen simulation length, so we do not report these in the results below.

The number of components is chosen in the range $N \in \{1, 2, 3, 4\}$, since with this choice of Δ, the maximum number of components, for which we can still solve the problems in a reasonable amount of time is four. The $N = 4$ CBMDPs solved in Figure 4 have $|S| = 83521$ and take between to 2 and 31 hours to solve, while the corresponding TBMDPs have $|S|$ between 1.2×10^6 and 1.7×10^8 and take between 17 and 109 hours to solve.

Finally, for the stochastic dependence, we test four different values for the Clayton-Lévy copula parameter, $\theta \in \{0, 0.2, 1.5, 3.0\}$, where $\theta = 0$ indicates that components are independent.

4.2. Single-component system

When $N = 1$, the setup cost, c_s, and stochastic dependence, θ, are redundant, and the system simplifies to the base model considered in de Jonge et al. (2017). The optimal policies are in this case the one-dimensional control-limit policies, AR and PIR. Figure 1 shows the difference in cost rate, $\tilde{g}$, when varying σ and c_b. The replacement cost is determined by $c_r = -1 - c_b$ to normalize g_{CM} to one. The largest difference between CBM and TBM is found where the breakdown cost, c_b, is comparatively large to the replacement cost c_r, and overall the benefit of CBM over TBM diminishes for both low and high values of σ. At σ close to zero both g_{CBM} and g_{TBM} are close to c_r since replacement can be made just in time before failure. At the other end for large σ the hazard rate becomes constant in time. It is therefore favorable to wait for a failure before replacing

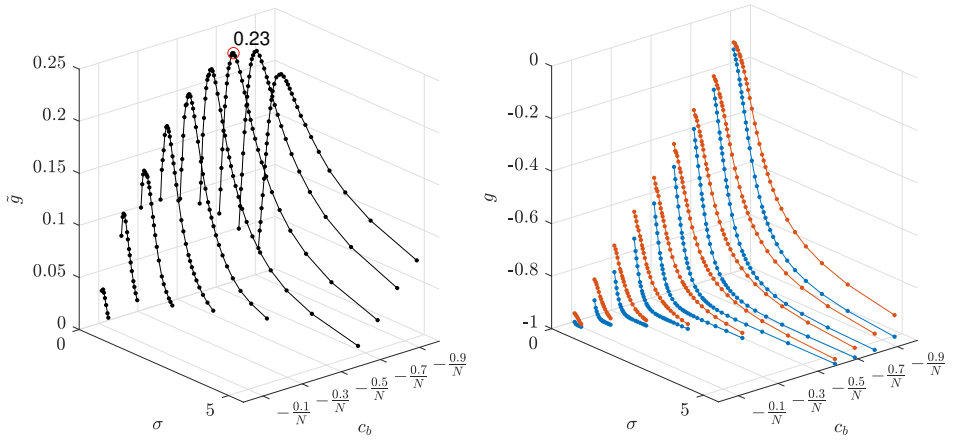


Figure 1: The difference in cost rates, $\tilde{g}$ (left), and the corresponding cost rates (right) for CBM, g_{CBM} (red), and TBM, g_{TBM} (blue), in the $N = 1$ system. Maximum $\tilde{g}$ is indicated with a red circle.

the component, because the hazard rate is the same immediately after the replacement, hence g_{CBM} and g_{TBM} are both close to $c_r + c_b = -1$ for high σ. A consequence of this is that for some combinations of σ and c_b the TBMDP state space truncation, D, is too low because the optimal replacement threshold lies at some higher age. In these cases the optimal policy is not identified, although it is closely approximated since the selected truncation point, D, defined above ensures that components rarely survive long enough to reach the true optimal replacement threshold.

Figure 1 mainly serves as a reference for the multi-component system results presented below. In all figures, we omit the data points, where the TBMDP truncation, D, is too low and $g_{\text{TBM}} < -0.99$ in order to make the figures easier to read and since these points do not reveal anything of significance.

4.3. Setup cost

In this section, we consider the $N = 2$ system with independent component deterioration ($\theta = 0$). Figure 2 shows $\tilde{g}$ wit varying setup cost, c_s. When $c_s = 0$ there is no economic dependence between the two components and the plot of the cost rate difference, $\tilde{g}$, is identical to the $N = 1$ system depicted in Figure 1. Comparing the maximum values of $\tilde{g}$ indicated in Figures 1 and 2 we see that $\tilde{g}$ becomes smaller in systems with a nonzero setup cost, and Figure 2 shows that the decrease is greater when the setup cost is larger.

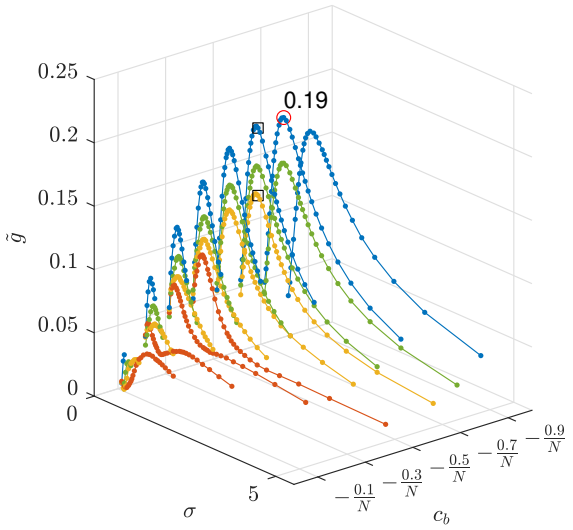
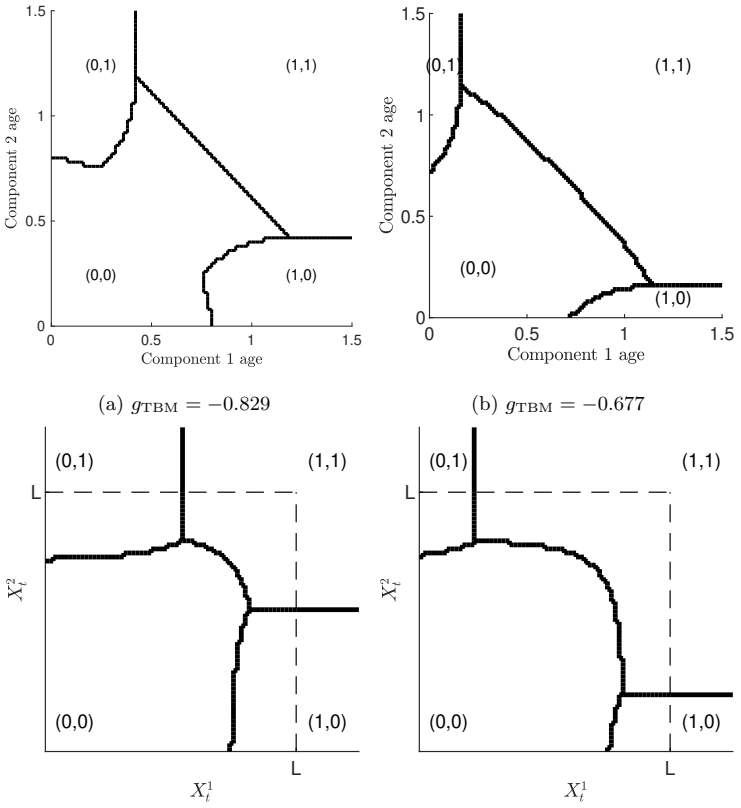


Figure 2: The difference in cost rate, $\tilde{g}$, for CBM and TBM in an $N = 2$ system with $\theta = 0$ and setup cost: $c_s = -0.1/N$ (blue), $c_s = -0.2/N$ (green), $c_s = -0.3/N$ (yellow), and $c_s = -0.5/N$ (red). Note that high values of c_s, means fewer feasible values of c_b, because the cost parameter combinations should comply with Equation (10). Maximum $\tilde{g}$ is indicated with a red circle. The optimal policies, π_{TBM}, π_{CBM} corresponding to the black squares are shown in Figure 3.

For a fixed breakdown cost c_b, both g_{TBM} and g_{CBM} improve when c_s becomes larger relative to c_r. Figure 3 shows the optimal policies, π_{TBM} and π_{CBM}, for a fixed value of $c_b = -0.6/N$ and two different values of c_s. g_{TBM} improves by 0.152 and g_{CBM} improves by 0.098 when c_s increases from $-0.1/N$ to $-0.3/N$. The reason for the improvements is that there are more system states where it makes sense to perform joint replacements, since the additional cost of a preventive replacement of one component if the other is being replaced is c_r, which correspondingly reduces from $-0.3/N$ to $-0.1/N$. It is evident from the cost rates in Figure 3 that g_{TBM} is more sensitive to c_s than g_{CBM}, which is why the difference, $\tilde{g}$, decreases as c_s increases. It is difficult to find an intuitive explanation of why this is the case, and therefore these results demonstrate the necessity of numerical experimentation when comparing CBM and TBM in multi-component systems.

4.4. Number of components

Figure 4 shows $\tilde{g}$ for systems with different number of components. For the small setup cost, $c_s = -0.1/N$, there is almost no visible changes in $\tilde{g}$ for different values of N. However, for



(a) $g_{\text{TBM}} = -0.829$

(b) $g_{\text{TBM}} = -0.677$

(c) $g_{\text{CBM}} = -0.645$

(d) $g_{\text{CBM}} = -0.547$

Figure 3: Optimal TBM policies (top), CBM policies (bottom), for $N = 2$, $\theta = 0$, $c_b = -0.6/N$, $\alpha = 4.0$, and $\beta = 3.46$. The setup and replacement costs are $c_s = -0.1/N$, $c_r = -0.3/N$ (left) and $c_s = -0.3/N$, $c_r = -0.1/N$ (right). The optimal action in each region is indicated with parenthesis.

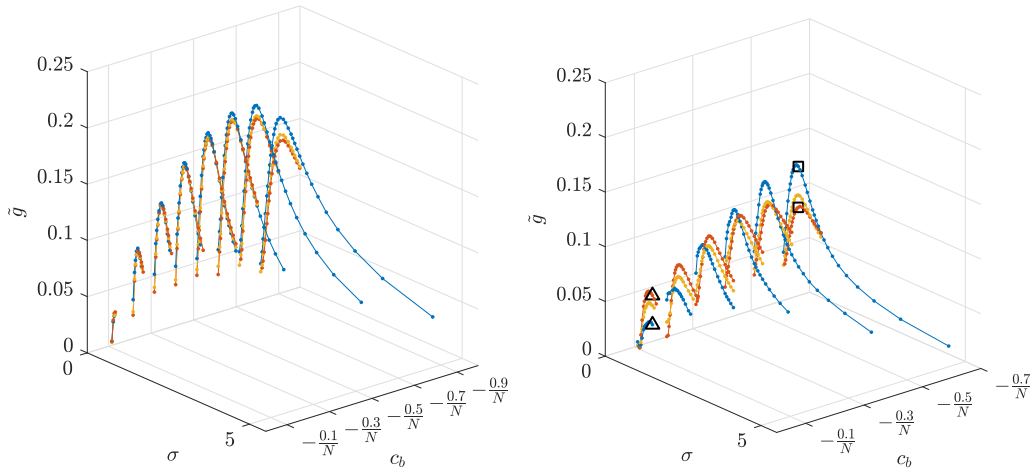


Figure 4: The difference in cost rate, $\tilde{g}$, for CBM and TBM in a system with $\theta = 0$ and with setup cost $c_s = -0.1/N$ (left), $c_s = -0.3/N$ (right) for $N = 2$ (blue), $N = 3$ (yellow), and $N = 4$ (red) components. The value of $\tilde{g}$ at the points marked with △ and □ are given in Table 2

Figure 4 mark	c_b	N	g_{TBM}	g_{CBM}	$\tilde{g}$
□	$-0.6/N$	2	-0.677	-0.547	0.130
		4	-0.560	-0.467	0.093
△	$-0.1/N$	2	-0.988	-0.960	0.028
		4	-0.980	-0.926	0.054

Table 2: Cost rates for varying number of components. $c_s = -0.3/N$, $\alpha = 4$, $\beta = 3.46$, $\theta = 0$.

$c_s = -0.3/N$ the cost rate difference, $\tilde{g}$, appear to become less sensitive to changes in c_b and c_r, as N increases. Indeed, $\tilde{g}$ is not always lower for $N = 4$ than $N = 2$. Table 2 shows the specific g_{TBM} and g_{CBM} values for two points in Figure 4, one where $N = 4$ has the highest $\tilde{g}$ value (△) and another where $N = 4$ has the lowest $\tilde{g}$ value (□).

In Table 2 and in general, the $N = 4$ system has better cost rates, g_{CBM} and g_{TBM}, than the $N = 2$ system. This demonstrates an interaction effect between c_s and N. Loosely explained, in an $N = 2$ system, the opportunities for joint replacement "saves" one setup cost, however in systems with a higher number of components, certain system states create opportunity for joint replacement of more than two component, which saves a multiple of the setup cost.

Figure 5 mark	c_s	c_b	θ	g_{TBM}	g_{CBM}	$\tilde{g}$
$\square$	$-0.3/N$	$-0.6/N$	0	-0.677	-0.547	0.130
			3.0	-0.628	-0.471	0.157
$\triangle$	0	$-0.6/N$	0	-0.880	-0.664	0.216
			3.0	-0.873	-0.664	0.209

Table 3: Cost rates for varying levels of stochastic dependence. $c_s = -0.3/N$, $\alpha = 4$, $\beta = 3.46$, $N = 2$.

4.5. Stochastic dependence

In this section we consider the $N = 2$ system with varying levels of stochastic dependence. Figure 5 shows $\tilde{g}$ for two different values of c_s. The $c_s = -0.3/N$ plot shows that $\tilde{g}$ is generally increases with θ. From the specific cost rates shown in Table 3, the increase in $\tilde{g}$ is because g_{CBM} improves more than g_{TBM}. The fact that both g_{CBM} and g_{TBM} improve can be explained by the deterioration processes X_t^1 and X_t^2 attaining a level of synchronization when θ increases, which leads to more opportunities for joint replacements.

For the case of $c_s = 0$, Figure 5 (left), the $N = 2$ system has no economic dependence, yet there is still a change in $\tilde{g}$ as θ increases. The CBM policy π_{CBM} remains the same and the cost rate g_{CBM} is identical to that of the $N = 1$ system in Figure 1 for both $\theta = 0$ and $\theta = 3$. However, as Figure 6 shows the optimal TBM policy π_{TBM} goes from being two independent AR control-limit policies at $\theta = 0$ to having a non-trivial structure at $\theta = 3$. This is somewhat surprising, although the change in g_{TBM} is only 0.007. In summary, $\tilde{g}$ generally increases with θ except for systems with a very small or no setup cost, for which $\tilde{g}$ decreases slightly as θ increases.

In systems with considerable setup cost, the increase in $\tilde{g}$ from a large θ does not outweigh the decrease in $\tilde{g}$ from the setup cost. That is, systems with no economic dependence have a larger $\tilde{g}$ than systems with both economic and stochastic dependence.

5. Conclusion

In this paper, we present a comparison between optimal TBM and CBM policies in a multi-component system. Previous studies that compare TBM and CBM only investigate single-component systems and heuristic policies. The general maintenance literature is increasingly focusing on modeling complex multi-component systems and developing CBM policies for such systems. The ana-

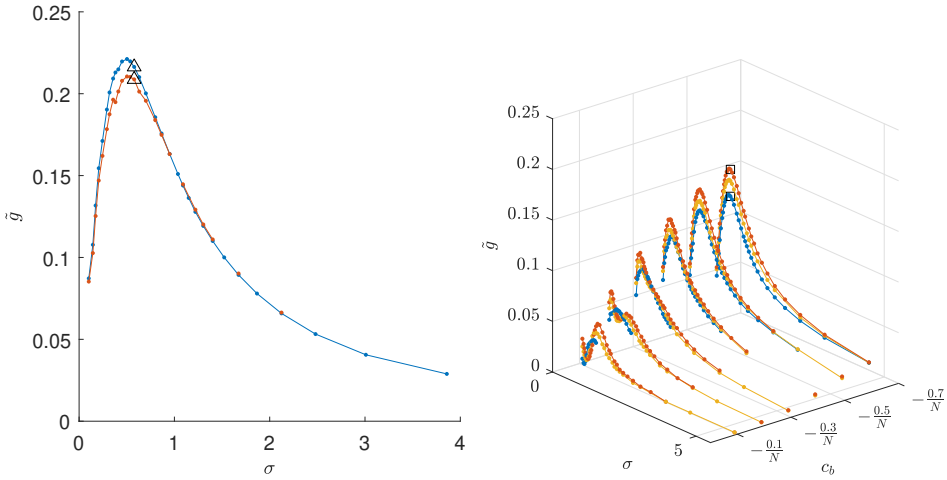
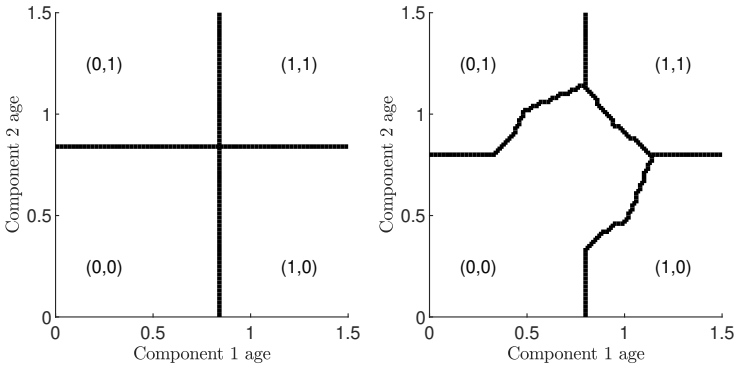


Figure 5: The difference in cost rate, $\tilde{g}$, for a system with $N = 2$, $c_s = 0$, $c_b = -0.6/N$ (left), $c_s = -0.3/N$ (right) for $\theta = 0$ (blue), $\theta = 1.5$ (yellow), and $\theta = 3.0$ (red). A weak dependence of $\theta = 0.2$ was also tested, but the resulting $\tilde{g}$ is indistinguishable from the $\theta = 0$ case on the plot. The value of $\tilde{g}$ at the points marked with $\triangle$ and $\square$ are given in Table 3



(a) $g_{\text{TBM}} = -0.880$ (b) $g_{\text{TBM}} = -0.873$

Figure 6: Optimal TBM policies for $\triangle$ marked points in Figure 5, which have parameters $N = 2$, $c_s = 0$, $c_b = -0.6/N$, $c_r = -0.4/N$ $\alpha = 4.0$, $\beta = 3.46$. $\theta = 0$ (left) and $\theta = 3$ (right). The optimal action in each region is indicated with parenthesis.

lysis we present is therefore an investigation into the underlying justification for this shift in the literature.

In the multi-component system we consider, there is economic dependence between components through a joint setup cost for maintenance actions and stochastic dependence modeled by dependent component deterioration increments. The degree of both of these dependences and the number of components in the system all affect the relative benefit of CBM over TBM. We measure the benefit as the difference between the cost rates of the optimal CBM policy and the optimal TBM policy.

When the setup cost increases, the cost rate difference decreases. A special case of the multi-component system is the single-component system, in which there is no setup cost. Consequently, the benefit of CBM over TBM is in general lower in the multi-component system compared to the single-component system. The benefit increases with the degree of stochastic dependence between components. However, a decrease in the cost rate difference as a result of a small setup cost outweighs the increase from stochastic dependence, even if this is a strong degree of stochastic dependence. As the number of components increase, the cost rate difference becomes less sensitive to the values of the replacement and breakdown cost parameters. Furthermore, both the CBM cost rate and the TBM cost rate generally improve as the number of components in the system increases.

In order to limit the number of variables in our sensitivity analysis, we do not include any cost for periodic inspection of the condition of the components in the CBM model. Therefore, the difference in cost rate, which we use as a measure to compare TBM and CBM, is equal to the inspection cost per time unit that would render the TBM and CBM policies equally good in terms of total costs. Considering that condition monitoring in practice involves a running inspection cost, the relative benefit of CBM over TBM is in fact smaller than the results from our numerical experiments suggest. Besides the inspection costs associated with CBM, the benefit of CBM over TBM is also determined by other practical circumstances (de Jonge et al., 2017). For a given real-world system, whether and how the currently used TBM policy or the future CBM policy are optimized also impacts the potential cost reduction from implementing CBM. Our comparative study is based on optimal policies identified using DP. An example of a policy, which is suboptimal but easier to optimize and implement in practice, is the (m_i, M_i) policy described in Section 1. A

comparison of TBM and CBM based on the (m_i, M_i) policy or another heuristic policy is a possible continuation of the research we present in this paper.

We limited the scope of this study, by only considering multi-component systems with economic dependence and stochastic dependence. Another possible continuation of this study, is to compare CBM and TBM in multi-component systems with structural dependence, for instance, by including systems with a reliability structure such as series-, parallel-, or K-out-of-N systems.

References

Abdel-Hameed, M. (1987). Inspection and maintenance policies of devices subject to deterioration. *Advances in Applied Probability*, *19*, 917–931. doi:10.2307/1427108.

Alaswad, S., & Xiang, Y. (2017). A review on condition-based maintenance optimization models for stochastically deteriorating system. *Reliability Engineering and System Safety*, *157*, 54–63. doi:10.1016/j.ress.2016.08.009.

Andersen, J. F., Andersen, A. R., Kulahci, M., & Nielsen, B. F. (2021). A numerical study of Markov decision process algorithms for multi-component replacement problems. *European Journal of Operational Research*, . doi:10.1016/J.EJOR.2021.07.007.

Barlow, R. E., & Proschan, F. (1996). *Mathematical Theory of Reliability*. SIAM.

Bouvard, K., Artus, S., Bérenguer, C., & Cocquempot, V. (2011). Condition-based dynamic maintenance operations planning & grouping. Application to commercial heavy vehicles. *Reliability Engineering and System Safety*, *96*, 601–610. doi:10.1016/j.ress.2010.11.009.

uit het Broek, M. A., Teunter, R. H., De Jonge, B., & Veldman, J. (2021). Joint condition-based maintenance and condition-based production optimization. *Reliability Engineering and System Safety*, *214*, 107743. doi:10.1016/j.ress.2021.107743.

Castro, I. T., Basten, R. J., & van Houtum, G. J. (2020). Maintenance cost evaluation for heterogeneous complex systems under continuous monitoring. *Reliability Engineering and System Safety*, *200*, 106745. doi:10.1016/j.ress.2019.106745.

Cherkaoui, H., Huynh, K. T., & Grall, A. (2018). Quantitative assessments of performance and robustness of maintenance policies for stochastically deteriorating production systems. *International Journal of Production Research*, *56*, 1089–1108. doi:10.1080/00207543.2017.1370563.

Dekker, R. (1996). Applications of maintenance optimization models: a review and analysis. *Reliability Engineering & System Safety*, *51*, 229–240. doi:10.1016/0951-8320(95)00076-3.

Dekker, R., Wildeman, R., & der Duyn Schouten, F. A. (1997). A review of multi-component maintenance models with economic dependence. *Mathematical methods of operations research*, *45*, 411–435. doi:10.1007/BF01194788.

Derman, C. (1963). On Optimal Replacement Rules when Changes of State are Markovian. In R. Bellman (Ed.), *Mathematical Optimization Techniques* (pp. 201–210). University of California Press Berkeley, CA.

Grall, A., Dieulle, L., Bérenguer, C., & Roussignol, M. (2002). Continuous-time predictive-maintenance scheduling for a deteriorating system. *IEEE Transactions on Reliability*, *51*, 141–150. doi:10.1109/TR.2002.1011518.

Grothe, O., & Hofert, M. (2015). Construction and sampling of Archimedean and nested Archimedean Lévy copulas. *Journal of Multivariate Analysis*, *138*, 182–198. doi:10.1016/j.jmva.2014.12.004.

Haurie, A., & L'Ecuyer, P. (1982). A Stochastic Control Approach to Group Preventive Replacement in a Multicomponent System. *IEEE Transactions on Automatic Control*, *27*, 387–393. doi:10.1109/TAC.1982.1102919.

Jiang, W., Hong, H. P., & Ren, J. (2021). Estimation of model parameters of dependent processes constructed using Lévy Copulas. *Communications in Statistics - Simulation and Computation*, *50*, 691–707. doi:10.1080/03610918.2019.1565581.

de Jonge, B., & Scarf, P. A. (2020). A review on maintenance optimization. *European Journal of Operational Research*, *285*, 805–824. doi:10.1016/j.ejor.2019.09.047.

de Jonge, B., Teunter, R., & Tinga, T. (2017). The influence of practical factors on the benefits of condition-based maintenance over time-based maintenance. *Reliability Engineering and System Safety*, *158*, 21–30. doi:10.1016/j.ress.2016.10.002.

Li, H., Deloux, E., & Dieulle, L. (2016). A condition-based maintenance policy for multi-component systems with Lévy copulas dependence. *Reliability Engineering & System Safety*, *149*, 44–55. doi:10.1016/j.ress.2015.12.011.

Nicolai, R. P., & Dekker, R. (2008). Optimal maintenance of multi-component systems: a review. In K. A. H. Kobbacy, & D. N. P. Murthy (Eds.), *Complex system maintenance handbook. Springer Series in Reliability Engineering* chapter 11. (pp. 263–286). Springer. doi:10.1007/978-1-84800-011-7_11.

van Noortwijk, J. M. (2009). A survey of the application of gamma processes in maintenance. *Reliability Engineering and System Safety*, *94*, 2–21. doi:10.1016/j.ress.2007.03.019.

Olde Keizer, M. C., Teunter, R. H., & Veldman, J. (2016). Clustering condition-based maintenance for systems with redundancy and economic dependencies. *European Journal of Operational Research*, *251*, 531–540. doi:10.1016/j.ejor.2015.11.008.

Özekici, S. (1988). Optimal periodic replacement of multicomponent reliability systems. *Operations Research*, *36*, 542–552. doi:10.1287/opre.36.4.542.

Park, K. S. (1988a). Optimal Continuous-Wear Limit Replacement under Periodic Inspections. *IEEE Transactions on Reliability*, *37*, 97–102. doi:10.1109/24.3722.

Park, K. S. (1988b). Optimal Wear-Limit Replacement with Wear-Dependent Failures. *IEEE Transactions on Reliability*, *37*, 293–294. doi:10.1109/24.3757.

Puterman, M. L. (2005). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. (2nd ed.). Wiley. doi:10.1002/9780470316887.

Shi, Y., Feng, Q., Shu, Y., & Xiang, Y. (2020). Multi-dimensional Lévy processes with Lévy copulas for multiple dependent degradation processes in lifetime analysis. *Quality Engineering*, *32*, 434–448. doi:10.1080/08982112.2020.1757704.

Shin, J. H., & Jun, H. B. (2015). On condition based maintenance policy. *Journal of Computational Design and Engineering*, *2*, 119–127. doi:10.1016/j.jcde.2014.12.006.

Sun, Q., Ye, Z.-S., & Chen, N. (2018). Optimal Inspection and Replacement Policies for Multi-Unit Systems Subject to Degradation. *IEEE Transactions on Reliability*, *67*, 401–413. doi:10.1109/TR.2017.2778283.

Wang, H. (2002). A survey of maintenance policies of deteriorating systems. *European Journal of Operational*

Research, *139*, 469–489. doi:10.1016/S0377-2217(01)00197-7.

Xu, J., Liang, Z., Li, Y.-F., & Wang, K. (2021). Generalized condition-based maintenance optimization for multi-component systems considering stochastic dependency and imperfect maintenance. *Reliability Engineering & System Safety*, *211*, 107592. doi:10.1016/j.ress.2021.107592.

Zuckerman, D. (1978). Optimal replacement policy for the case where the damage process is a one-sided Lévy process. *Stochastic Processes and their Applications*, *7*, 141–151. doi:10.1016/0304-4149(78)90013-3.

CHAPTER 6

Summary and Perspectives

The main topic of this dissertation is modeling and optimization of maintenance in multi-component systems. Modeling the dynamics of a given real-world multi-component system is challenging in itself, and from the review we presented in Chapter 2 it is clear that there are many aspects to consider. Moreover, the possible dependences between components in such systems lead to some very difficult optimization problems.

The underlying motivation for the research presented in Chapter 3 was the question of the practicality of DP optimization compared to other forms of optimization, given the general trade-off between computational requirements and the performance of the resulting policy. DP optimization generally results in the best policy, but the computational requirements grow exponentially with the number of components in the system. We considered a generic multi-component system and investigated the practical system size limit in terms of computation in DP algorithms, both in the case of TBM and CBM. A feature of the chosen system is that the components deteriorate according to a stochastic process that is continuous in time and state space, namely the gamma process. This allowed us to explore the effects of discretization in the CBM case, which is a necessary step when using iterative DP algorithms. Compared to results presented in other studies that consider st systems and DP optimization, the results of our numerical experiments indicate that relatively few discretization intervals are needed to obtain a near-optimal policy.

Chapter 4 is a more detailed investigation of the possible ways of performing the discretization in the CBM model. The conclusion from this study is that the choice of discretization procedure affects how accurate the DP algorithm estimates the value of the output policy, but that the policy itself is fairly robust to the choice of discretization method even when the discretization is coarse. Among the different methods of discretization we investigated, one is only applicable when the components deteriorate independently of each other. This particular method is based on the expected number of transitions between discretization intervals observed in the continuous deterioration process. This method proved to produce a more accurate estimate of the value of the output policy. We demonstrated how to exploit this in a

multigrid algorithm in order to speed up the convergence of the DP algorithm.

When the CBM model becomes so large that DP is no longer a feasible option, approximate algorithms may be used. Most ADP and RL algorithms are based on Monte-Carlo simulation of the system, and for this purpose we may use the original continuous deterioration process. In this case, the discretization is not strictly needed. Nonetheless, for computational reasons it might be more practical to use the discretized process for simulating of the system. For instance, the deterioration of the stochastically dependent components we consider in Chapters 3 and 5 is modeled by a multivariate gamma process with the Clayton-Lévy copula dependence. When the components are weekly dependent, simulating one time step in this process accurately requires a sum of 4000 sampled jump sizes, and this can potentially make an approximate algorithm quite slow. In that sense, our results related to discretization have a broader relevance than iterative DP algorithms.

Besides being a technical detail in the DP optimization, the discretization also has a practical interpretation. In CBM the condition monitoring is sometimes performed by inspecting the system and classifying the observed deterioration on a qualitatively labeled scale with a small number of levels. The discretization method in Chapter 4 based on the expected number of transitions between discretization intervals is conceptually the same as modeling the deterioration as a DTMC with a state for each qualitative level. Any maintenance optimization based on a simple deterioration model such as this could result in a policy that performs poorly. However, in the example we consider with the gamma process to describe deterioration, a coarse discretization is shown to be sufficient, and this result is therefore a validation of the condition monitoring practice described above.

In Chapter 5 we compared TBM and CBM in a multi-component system. Modern technical systems are often comprised of several components, and within industry there is a growing interest in CBM. The work in Chapter 5 is a quantification of how much can be gained by implementing CBM in a multi-component system. Designing the numerical experiments that help answer this question is not as straightforward as for a single-component system. There are more choices for the candidate multi-component system model because of the dependences between components. Furthermore, the optimization of TBM and CBM policies can be done in different ways, and the sensitivity analysis is computationally demanding when many system parameter settings are considered. We speculate the latter is the primary reason why a similar study was not carried out before. In our study, we reuse the model framework developed in Chapter 3 and investigate how the optimal CBM policy and the optimal TBM policy differ in performance when the parameters of the multi-component system are varied. The parameters that are exclusive to the multi-component version of the considered system are the setup cost, the number of components, and the degree of stochastic dependence between components. We observed that increasing the setup cost decreases the difference between CBM and TBM, while increasing the degree of stochastic dependence increases the difference. When the number of components increases, the sensitivity to the ratio between PM cost and CM cost decreases. Finally, of all the possible combinations of system parameter values, most result in a

smaller difference between CBM and TBM than the corresponding difference in the single-component system.

6.1 Research directions

From the conclusions and insights we have obtained through the work in the previous chapters, we see a number of possible topics for further research.

Continuous-time discretization In all the systems we have considered, the deterioration of components has been assumed to be continuous in time and space. In the CBM models we then discretized this process into a DTMC. The resulting MDP has equidistant epochs corresponding to periodic inspection of the system condition. However, the primary motivation for the discretization is to obtain a finite state space, because this enables the use of DP for optimization. Therefore, an alternative method of discretization is to use a CTMC, which results in an SMDP model. The time to failure of a component in the discretized deterioration process is then phase-type distributed. The class of phase-type distributions is dense in distributions with positive support, so the CTMC is in fact a sensible choice for approximating the time to failure of a continuous deterioration process, such as a gamma process. It is therefore also plausible, that the CTMC discretization is a better choice than the DTMC in terms the number of discretization intervals needed for obtaining a near-optimal policy. The practical interpretation of the CBM policy obtained from the SMDP model is also somewhat different. The decision epochs in the SMDP are not equidistant, so the interpretation of periodic inspection of the system is not valid. Instead, the policy obtained from the SMDP is useful in modeling a system with continuous monitoring, where it is possible to perform maintenance actions at any given moment.

Harmonization In Section 2.3.4, we reviewed some results from the literature that compared the performance of heuristics to that of optimal policies. It is not obvious from these results that we can draw any general conclusions for which kind of systems a heuristic performs nearly as well as the optimal policy. However, the number of components is a factor that partially determines this in multi-component systems with economic dependence through a joint setup cost, such as the systems considered in this dissertation. When a multi-component system has many components, then at almost all decision epochs, some components will have failed and require maintenance. In this case, deciding if maintenance activities for different components should be grouped is less important, since the setup cost is paid at almost every epoch anyway. This effect is referred to as harmonization in Dekker et al. (1996). For such systems, an optimization problem, where all components are considered simultaneously, is not necessarily the best approach. Instead we can optimize a policy for each individual component separately, which is the basis for the heuristic developed in

Dekker et al. (1996). Another example of this approach is Zhu et al. (2015), in which a numerical example with 60 components of three different types is considered. The three types have expected times to failure of 116, 141, and 143 days, respectively, while the optimal inspection interval is found to be 36 days. For this system and this inspection policy it is demonstrated in the paper that every epoch is likely to have a failed component. However, the choice of inspection interval affects the degree of harmonization. For instance, if the 60 components are inspected every day, the fraction of epochs, where no failure has occurred, will be larger.

As far as we know, no systematic study of the harmonization effect has been conducted so far. This is an interesting research subject, because a better understanding of this effect will help to determine the appropriate method of optimization a given multi-component system. In systems with a relatively small number of components, DP algorithms can be used to find optimal maintenance policies. For systems with a very large number components, heuristics that are easy to optimize may perform almost as well as the optimal policy due to the harmonization effect, so there is no reason to pursue the latter. It is not obvious whether there is a gap between small systems, where we can obtain the optimal policy, and large systems, where we no longer need it. If the gap exists, it is also an interesting question whether approximate algorithms from ADP/RL are enough to deal with these systems of intermediate size.

Case studies All the models we have considered in the previous chapters are of generic multi-component systems. Although no specific real-world system has been modeled, the motivation behind the numerical experiments we have conducted was to test the applicability of MDP and DP. Undoubtedly, the ultimate test of the applicability of maintenance optimization is to use the methodology on real-world technical systems.

Maintenance is in principle an applied research field, but in spite of this relatively few case studies are being published in maintenance-related journals. In Dekker (1996), the author states that even though maintenance optimization has thrived as a mathematical discipline within operations research, its impact on decision making in maintenance organizations has been limited. Another paper that makes this point is Scarf (1997), where the overall purpose of the paper is to be an appeal for maintenance researchers to do more collaborations with industry. More recently, Fraser et al. (2015) quantify the problem by analyzing how many maintenance papers in the top maintenance journals show any empirical evidence for their models being of practical use. Between the years 1995 and 2010 the three most prominent maintenance journals, *Reliability Engineering and System Safety*, *Journal of Quality in Maintenance Engineering*, and *International Journal of Quality and Reliability Management* published 493 papers on maintenance models, and only 39 of them contained empirical evidence, which corresponds to 8%. This rate is arguably too low for an applied field such as maintenance, but the authors acknowledge that a comparison with the rate of empirical studies in other fields is needed. Of all the references we mention in this

thesis that are published after 2010, the majority are purely theoretical, so the issue is still present today.

It is not surprising to us that there is a lack of published case studies. Since a successful case study of maintenance optimization is a collaboration project that requires a high level of commitment from both the researchers and the industry partner, the research outcome can be more unpredictable and work may take longer than theoretical research (Scarf 1997). Furthermore, assessing the success of implementing a new maintenance policy should take a long time if the policy is optimized for a long horizon. If the change of policy is successful, another issue is that private companies may be reluctant to publish the results of such a study, if it means that they lose a competitive edge (Dekker 1996). Nonetheless, from a perspective of advancing the field of maintenance, case studies are worthwhile the effort, because they bridge the gap between theory and practice that is currently too large.

Bibliography

13848-5, E. (2017). *Railway applications - Track - Track geometry quality - Part 5: Geometric quality levels*.

Abdel-Hameed, M. (1987). "Inspection and maintenance policies of devices subject to deterioration." In: *Advances in Applied Probability* 19.4, pp. 917–931. DOI: 10.2307/1427108.

Ahmad, R. and Kamaruddin, S. (2012). "An overview of time-based and condition-based maintenance in industrial application." In: *Computers and Industrial Engineering* 63.1, pp. 135–149. DOI: 10.1016/j.cie.2012.02.002.

Alaswad, S. and Xiang, Y. (2017). "A review on condition-based maintenance optimization models for stochastically deteriorating system." In: *Reliability Engineering and System Safety* 157, pp. 54–63. DOI: 10.1016/j.ress.2016.08.009.

Almeida, A. T. de, Ferreira, R. J. P., and Cavalcante, C. A. V. (2015). "A review of the use of multicriteria and multi-objective models in maintenance and reliability." In: *IMA Journal of Management Mathematics* 26.3, pp. 249–271. DOI: 10.1093/IMAMAN/DPV010.

Altman, E. (1999). *Constrained Markov decision processes*. CRC press.

Andriotis, C. P. and Papakonstantinou, K. G. (2019). "Managing engineering systems with large state and action spaces through deep reinforcement learning." In: *Reliability Engineering & System Safety* 191, p. 106483. DOI: 10.1016/j.ress.2019.04.036.

Andrychowicz, M. et al. (2020). "Learning dexterous in-hand manipulation." In: *International Journal of Robotics Research* 39.1, pp. 3–20. DOI: 10.1177/0278364919887447.

Archibald, T. W. and Dekker, R. (1996). "Modified block-replacement for multiple-component systems." In: *IEEE Transactions on Reliability* 45.1, pp. 75–83. DOI: 10.1109/24.488920.

Arismendi, R., Barros, A., and Grall, A. (2021). "Piecewise deterministic Markov process for condition-based maintenance models — Application to critical infrastructures with discrete-state deterioration." In: *Reliability Engineering and System Safety* 212, p. 107540. DOI: 10.1016/j.ress.2021.107540.

Assaf, D. and Shanthikumar, J. G. (1987). "Optimal Group Maintenance Policies with Continuous and Periodic Inspections." In: *Management Science* 33.11, pp. 1367–1509. DOI: 10.1287/MNSC.33.11.1440.

Barlow, R. E. and Proschan, F. (1996). *Mathematical Theory of Reliability*. SIAM.

Barreto, A. M. S., Pineau, J., and Precup, D. (2014). "Policy Iteration Based on Stochastic Factorization." In: *Journal of Artificial Intelligence Research* 50, pp. 763–803. DOI: 10.1613/jair.4301.

Bertsekas, D. P. (1975). "Convergence of Discretization Procedures in Dynamic Programming." In: *IEEE Transactions on Automatic Control* 20.3, pp. 415–419. DOI: 10.1109/TAC.1975.1100984.

Bertsekas, D. P. (2012). *Dynamic Programming and Optimal Control: Vol. 2*. 4th ed. Athena scientific.

Bevilacqua, M. and Braglia, M. (2000). "Analytic hierarchy process applied to maintenance strategy selection." In: *Reliability Engineering and System Safety* 70.1, pp. 71–83. DOI: 10.1016/S0951-8320(00)00047-8.

Bian, L. and Gebraeel, N. (2014). "Stochastic modeling and real-time prognostics for multi-component systems with degradation rate interactions." In: *IIE Transactions* 46.5, pp. 470–482. DOI: 10.1080/0740817X.2013.812269.

Bouvard, K. et al. (2011). "Condition-based dynamic maintenance operations planning & grouping. Application to commercial heavy vehicles." In: *Reliability Engineering and System Safety* 96.6, pp. 601–610. DOI: 10.1016/j.ress.2010.11.009.

Bowman, R. A. and Schmee, J. (2001). "Pricing and Managing a Maintenance Contract for a Fleet of Aircraft Engines." In: *SIMULATION* 76.2, pp. 69–77. DOI: 10.1177/003754970107600202.

Broek, M. A. uit het et al. (2021). "Joint condition-based maintenance and condition-based production optimization." In: *Reliability Engineering and System Safety* 214, p. 107743. DOI: 10.1016/j.ress.2021.107743.

Byon, E. and Ding, Y. (2010). "Season-dependent condition-based maintenance for a wind turbine using a partially observed markov decision process." In: *IEEE Transactions on Power Systems* 25.4, pp. 1823–1834. DOI: 10.1109/TPWRS.2010.2043269.

Caetano, L. F. and Teixeira, P. F. (2015). "Optimisation model to schedule railway track renewal operations: a life-cycle cost approach." In: *Structure and Infrastructure Engineering* 11.11, pp. 1524–1536. DOI: 10.1080/15732479.2014.982133.

Castanier, B., Grall, A., and Bérenguer, C. (2005). "A condition-based maintenance policy with non-periodic inspections for a two-unit series system." In: *Reliability Engineering & System Safety* 87.1, pp. 109–120. DOI: 10.1016/j.ress.2004.04.013.

Castro, I. T., Basten, R. J., and Houtum, G. J. van (2020). "Maintenance cost evaluation for heterogeneous complex systems under continuous monitoring." In: *Reliability Engineering and System Safety* 200, p. 106745. DOI: 10.1016/j.ress.2019.106745.

Chan, G. K. and Asgarpoor, S. (2006). "Optimum maintenance policy with Markov processes." In: *Electric Power Systems Research* 76.6-7, pp. 452–456. DOI: 10.1016/J.EPSR.2005.09.010.

Chen, D. and Trivedi, K. S. (2005). "Optimization for condition-based maintenance with semi-Markov decision process." In: *Reliability Engineering & System Safety* 90.1, pp. 25–29. DOI: 10.1016/J.RESS.2004.11.001.

Chen, N. et al. (2015). "Condition-based maintenance using the inverse Gaussian degradation model." In: *European Journal of Operational Research* 243.1, pp. 190–199. DOI: 10.1016/j.ejor.2014.11.029.

Cherkaoui, H., Huynh, K. T., and Grall, A. (2018). "Quantitative assessments of performance and robustness of maintenance policies for stochastically deteriorating production systems." In: *International Journal of Production Research* 56.3, pp. 1089–1108. DOI: 10.1080/00207543.2017.1370563.

Cho, D. I. and Parlar, M. (1991). "A survey of maintenance models for multi-unit systems." In: *European Journal of Operational Research* 51.1, pp. 1–23. DOI: 10.1016/0377-2217(91)90141-H.

Chow, C.-S. and Tsitsiklis, J. N. (1991). "An optimal one-way multigrid algorithm for discrete-time stochastic control." In: *IEEE transactions on automatic control* 36.8, pp. 898–914. DOI: 10.1109/9.133184.

Dekker, R. (1996). "Applications of maintenance optimization models: a review and analysis." In: *Reliability Engineering & System Safety* 51.3, pp. 229–240. DOI: 10.1016/0951-8320(95)00076-3.

Dekker, R. and Scarf, P. A. (1998). "On the impact of optimisation models in maintenance decision making: The state of the art." In: *Reliability Engineering and System Safety* 60.2, pp. 111–119. DOI: 10.1016/S0951-8320(98)83004-4.

Dekker, R., Wildeman, R., and der Duyn Schouten, F. A. (1997). "A review of multi-component maintenance models with economic dependence." In: *Mathematical methods of operations research* 45.3, pp. 411–435. DOI: 10.1007/BF01194788.

Dekker, R., Wildeman, R., and Egmond, R. van (1996). "Joint replacement in an operational planning phase." In: *European Journal of Operational Research* 91.1, pp. 74–88. DOI: 10.1016/0377-2217(94)00363-7.

Deloux, E., Castanier, B., and Bérenguer, C. (2009). "Predictive maintenance policy for a gradually deteriorating system subject to stress." In: 94, pp. 418–431. DOI: 10.1016/j.ress.2008.04.002.

Derman, C. (1963). "On Optimal Replacement Rules when Changes of State are Markovian." In: *Mathematical Optimization Techniques*. Ed. by R. Bellman. University of California Press Berkeley, CA, pp. 201–210.

Dieulle, L., Grall, A., and Bérenguer, C. (2002). "A condition-based maintenance policy for stochastically deteriorating systems." In: *Reliability Engineering & System Safety* 76, pp. 167–180. DOI: 10.1016/S0951-8320(01)00148-X.

Ditlevsen, O. and Madsen, H. O. (1996). *Structural Reliability Methods*. 1st ed. John Wiley & Sons Ltd.

Eegholm-Larsen, J. and Olesen, V. (2021). "Scheduling of Railway Tamping using Markov Decision Processes." BSc thesis. Technical University of Denmark.

Elwany, A. H., Gebraeel, N. Z., and Maillart, L. M. (2011). "Structured Replacement Policies for Components with Complex Degradation Processes and Dedicated Sensors." In: *Operations research* 59.3, pp. 684–695. DOI: 10.1287/opre.1110.0912.

Eruguz, A. S., Tan, T., and Houtum, G. J. van (2017). "A survey of maintenance and service logistics management: Classification and research agenda from a maritime

sector perspective." In: *Computers and Operations Research* 85, pp. 184–205. DOI: 10.1016/j.cor.2017.03.003.

Feng, Q. et al. (2015). "Reliability and condition-based maintenance for multi-stent systems with stochastic-dependent competing risk processes." In: *The International Journal of Advanced Manufacturing Technology 2015 80:9* 80.9, pp. 2027–2040. DOI: 10.1007/S00170-015-7182-3.

Fraser, K., Hvolby, H. H., and Tseng, T. L. B. (2015). "Maintenance management models: a study of the published literature to identify empirical evidence: A greater practical focus is needed." In: *International Journal of Quality & Reliability Management* 32.6, pp. 635–664. DOI: 10.1108/IJQRM-11-2013-0185.

Golabi, K., Kulkarni, R. B., and Way, G. B. (1982). "A Statewide Pavement Management System." In: *INFORMS Journal on Applied Analytics* 12.6, pp. 5–21. DOI: 10.1287/inte.12.6.5.

Grall, A. et al. (2002). "Continuous-time predictive-maintenance scheduling for a deteriorating system." In: *IEEE Transactions on Reliability* 51.2, pp. 141–150. DOI: 10.1109/TR.2002.1011518.

Grothe, O. and Hofert, M. (2015). "Construction and sampling of Archimedean and nested Archimedean Lévy copulas." In: *Journal of Multivariate Analysis* 138, pp. 182–198. DOI: 10.1016/j.jmva.2014.12.004.

Guo, C. et al. (2013). "A maintenance optimization model for mission-oriented systems based on Wiener degradation." In: *Reliability Engineering & System Safety* 111, pp. 183–194. DOI: 10.1016/J.RESS.2012.10.015.

Hao, S., Yang, J., and Berenguer, C. (2020). "Condition-based maintenance with imperfect inspections for continuous degradation processes." In: *Applied Mathematical Modelling* 86, pp. 311–334. DOI: 10.1016/j.apm.2020.05.013.

Haurie, A. and L'Ecuyer, P. (1982). "A Stochastic Control Approach to Group Preventive Replacement in a Multicomponent System." In: *IEEE Transactions on Automatic Control* 27.2, pp. 387–393. DOI: 10.1109/TAC.1982.1102919.

Hong, H. P. et al. (2014). "Optimal condition-based maintenance decisions for systems with dependent stochastic degradation of components." In: *Reliability Engineering & System Safety* 121, pp. 276–288. DOI: 10.1016/J.RESS.2013.09.004.

Hontelez, J. A., Burger, H. H., and Wijnmalen, D. J. (1996). "Optimum condition-based maintenance policies for deteriorating systems with partial information." In: *Reliability Engineering & System Safety* 51.3, pp. 267–274. DOI: 10.1016/0951-8320(95)00087-9.

ITF (2021). "Transport infrastructure investment and maintenance." In: *International Transport Forum (database)* (accessed. DOI: 10.1787/g2g55573-en.

Jardine, A. K., Lin, D., and Banjevic, D. (2006). "A review on machinery diagnostics and prognostics implementing condition-based maintenance." In: *Mechanical Systems and Signal Processing* 20.7, pp. 1483–1510. DOI: 10.1016/J.YMSSP.2005.09.012.

Jiang, D. R. and Powell, W. B. (2015). "An Approximate Dynamic Programming Algorithm for Monotone Value Functions." In: *Operations Research* 63.6, pp. 1489–1511. DOI: 10.1287/opre.2015.1425.

Jiang, W., Hong, H. P., and Ren, J. (2021). "Estimation of model parameters of dependent processes constructed using Lévy Copulas." In: *Communications in Statistics - Simulation and Computation* 50.3, pp. 691–707. DOI: 10.1080/03610918.2019.1565581.

Jonge, B. de and Scarf, P. A. (2020). "A review on maintenance optimization." In: *European Journal of Operational Research* 285, pp. 805–824. DOI: 10.1016/j.ejor.2019.09.047.

Jonge, B. de, Teunter, R., and Tinga, T. (2017). "The influence of practical factors on the benefits of condition-based maintenance over time-based maintenance." In: *Reliability Engineering and System Safety* 158, pp. 21–30. DOI: 10.1016/j.ress.2016.10.002.

Kallsen, J. and Tankov, P. (2006). "Characterization of dependence of multidimensional Lévy processes using Lévy copulas." In: *Journal of Multivariate Analysis* 97.7, pp. 1551–1572. DOI: 10.1016/J.JMVA.2005.11.001.

Klutke, G. A. and Yang, Y. (2002). "The availability of inspected systems subject to shocks and graceful degradation." In: *IEEE Transactions on Reliability* 51.3, pp. 371–374. DOI: 10.1109/TR.2002.802891.

Kozin, F. and Bogdanoff, J. L. (1989). "Probabilistic models of fatigue crack growth: Results and speculations." In: *Nuclear Engineering and Design* 115.1, pp. 143–171. DOI: 10.1016/0029-5493(89)90267-7.

Kulahci, M. et al. (2020). "Experiences with big data: Accounts from a data scientist's perspective." In: *Quality Engineering* 32.4, pp. 529–542. DOI: 10.1080/08982112.2019.1686641.

Kurt, M. and Kharoufeh, J. P. (2010). "Monotone optimal replacement policies for a Markovian deteriorating system in a controllable environment." In: *Operations Research Letters* 38.4, pp. 273–279. DOI: 10.1016/j.orl.2010.03.001.

Li, H., Deloux, E., and Dieulle, L. (2016). "A condition-based maintenance policy for multi-component systems with Lévy copulas dependence." In: *Reliability Engineering & System Safety* 149, pp. 44–55. DOI: 10.1016/j.ress.2015.12.011.

Liu, Y., Chen, Y., and Jiang, T. (2018). "On sequence planning for selective maintenance of multi-state systems under stochastic maintenance durations." In: *European Journal of Operational Research* 268.1, pp. 113–127. DOI: 10.1016/J.EJOR.2017.12.036.

Liu, Y. and Huang, H. Z. (2010). "Optimization of multi-state elements replacement policy for multi-state systems." In: *Proceedings - Annual Reliability and Maintainability Symposium*. DOI: 10.1109/RAMS.2010.5448061.

Marseguerra, M., Zio, E., and Podofillini, L. (2002). "Condition-based maintenance optimization by means of genetic algorithms and Monte Carlo simulation." In: *Reliability Engineering & System Safety* 77.2, pp. 151–165. DOI: 10.1016/S0951-8320(02)00043-1.

Medury, A. and Madanat, S. (2013). "Incorporating network considerations into pavement management systems: A case for approximate dynamic programming." In: *Transportation Research Part C: Emerging Technologies* 33, pp. 134–150. DOI: 10.1016/j.trc.2013.03.003.

Meeker, W. Q. and Escobar, L. A. (2014). *Statistical methods for reliability data*. John Wiley & Sons, Inc.

Mercier, S. and Castro, I. T. (2013). "On the Modelling of Imperfect Repairs for a Continuously Monitored Gamma Wear Process Through Age Reduction." In: *Journal of Applied Probability* 50.4, pp. 1057–1076. DOI: 10.1239/JAP/1389370099.

Mercier, S. and Castro, I. T. (2019). "Stochastic comparisons of imperfect maintenance models for a gamma deteriorating system." In: *European Journal of Operational Research* 273.1, pp. 237–248. DOI: 10.1016/J.EJOR.2018.06.020.

Mnih, V. et al. (2015). "Human-level control through deep reinforcement learning." In: *Nature* 518.7540, pp. 529–533. DOI: 10.1038/nature14236.

Mobley, R. K. (2002). *An Introduction to Predictive Maintenance*. 2nd ed. Elsevier.

Moghaddam, K. S. and Usher, J. S. (2011). "Sensitivity analysis and comparison of algorithms in preventive maintenance and replacement scheduling optimization models." In: *Computers & Industrial Engineering* 61.1, pp. 64–75. DOI: 10.1016/J.CIE.2011.02.012.

Montgomery, D. C. and Woodall, W. H. (2008). "An overview of six sigma." In: *International Statistical Review* 76.3, pp. 329–346. DOI: 10.1111/j.1751-5823.2008.00061.x.

Morato, P. G. et al. (2019). "POMDP based Maintenance Optimization of Offshore Wind Substructures including Monitoring." In: *13th International Conference on Applications of Statistics and Probability in Civil Engineering(ICASP13)*. Seoul. DOI: 10.22725/ICASP13.067.

Nguyen, K. T. P. et al. (2019). "Joint optimization of monitoring quality and replacement decisions in condition-based maintenance." In: *Reliability Engineering & System Safety* 189, pp. 177–195. DOI: 10.1016/j.ress.2019.04.034.

Nguyen, K. A., Do, P., and Grall, A. (2015). "Multi-level predictive maintenance for multi-component systems." In: *Reliability Engineering & System Safety* 144, pp. 83–94. DOI: 10.1016/J.RESS.2015.07.017.

Nicolai, R. P. and Dekker, R. (2008). "Optimal maintenance of multi-component systems: a review." In: *Complex system maintenance handbook. Springer Series in Reliability Engineering*. Ed. by K. A. H. Kobbacy and D. N. P. Murthy. Springer. Chap. 11, pp. 263–286. DOI: 10.1007/978-1-84800-011-7_11.

Noortwijk, J. M. van (2009). "A survey of the application of gamma processes in maintenance." In: *Reliability Engineering and System Safety* 94.1, pp. 2–21. DOI: 10.1016/j.ress.2007.03.019.

Noortwijk, J. M. van, Kok, M., and Cooke, R. (1997). "Optimal Maintenance Decisions for the Sea-Bed Protection of the Eastern-Scheldt Barrier." In: *Engineering Probabilistic Design and Maintenance for Flood Protection*, pp. 25–56. DOI: 10.1007/978-1-4613-3397-5_2.

Noortwijk, J. M. van, Cooke, R. M., and Kok, M. (1995). "A Bayesian failure model based on isotropic deterioration." In: *European Journal of Operational Research* 82.2, pp. 270–282. DOI: 10.1016/0377-2217(94)00263-C.

Olde Keizer, M. C., Flapper, S. D. P., and Teunter, R. H. (2017). "Condition-based maintenance policies for systems with multiple dependent components: A review."

In: *European Journal of Operational Research* 261.2, pp. 405–420. DOI: 10.1016/j.ejor.2017.02.044.

Olde Keizer, M. C., Teunter, R. H., and Veldman, J. (2016). "Clustering condition-based maintenance for systems with redundancy and economic dependencies." In: *European Journal of Operational Research* 251.2, pp. 531–540. DOI: 10.1016/j.ejor.2015.11.008.

Olde Keizer, M. C. et al. (2018). "Condition-based maintenance for systems with economic dependence and load sharing." In: *International Journal of Production Economics* 195, pp. 319–327. DOI: 10.1016/j.ijpe.2017.10.030.

Özekici, S. (1988). "Optimal periodic replacement of multicomponent reliability systems." In: *Operations Research* 36.4, pp. 542–552. DOI: 10.1287/opre.36.4.542.

Papakonstantinou, K. G. and Shinozuka, M. (2014a). "Optimum inspection and maintenance policies for corroded structures using partially observable Markov decision processes and stochastic, physically based models." In: *Probabilistic Engineering Mechanics* 37, pp. 93–108. DOI: 10.1016/j.probengmech.2014.06.002.

Papakonstantinou, K. G. and Shinozuka, M. (2014b). "Planning structural inspection and maintenance policies via dynamic programming and Markov processes. Part II: POMDP implementation." In: *Reliability Engineering & System Safety* 130, pp. 214–224. DOI: https://doi.org/10.1016/j.ress.2014.04.006.

Pham, H. and Wang, H. (1996). "Imperfect maintenance." In: *European Journal of Operational Research* 94.3, pp. 425–438. DOI: 10.1016/S0377-2217(96)00099-9.

Pineau, J., Gordon, G., and Thrun, S. (2003). "Point-Based Value Iteration: An Anytime Algorithm for POMDPs." In: *Proceedings of the 18th International Joint Conference on Artificial Intelligence*. IJCAI'03. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 1025–1030.

Powell, W. B. (2011). *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. 2nd ed. John Wiley & Sons, Inc. DOI: 10.1002/9781118029176.

Puterman, M. L. (2005). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. 2nd ed. Wiley, p. 649. DOI: 10.1002/9780470316887.

Ranjkesh, S. H., Hamadani, A. Z., and Mahmoodi, S. (2019). "A new cumulative shock model with damage and inter-arrival time dependency." In: *Reliability Engineering & System Safety* 192, p. 106047. DOI: 10.1016/J.RESS.2018.01.006.

Rust, J. (1986). "When is it Optimal to Kill off the Market for Used Durable Goods?" In: *Econometrica* 54.1, p. 65. DOI: 10.2307/1914157.

Rust, J. (1997). "Using Randomization to Break the Curse of Dimensionality." In: *Econometrica* 65.3, pp. 487–516. DOI: 10.2307/2171751.

Scarf, P. A. (1997). "On the application of mathematical models in maintenance." In: *European Journal of Operational Research* 99.3, pp. 493–506. DOI: 10.1016/S0377-2217(96)00316-5.

Sedghi, M. et al. (2021). "A taxonomy of railway track maintenance planning and scheduling: A review and research trends." In: *Reliability Engineering & System Safety* 215, p. 107827. DOI: 10.1016/J.RESS.2021.107827.

Shafiee, M., Finkelstein, M., and Bérenguer, C. (2015). "An opportunistic condition-based maintenance policy for offshore wind turbine blades subjected to degrada-

tion and environmental shocks." In: *Reliability Engineering & System Safety* 142, pp. 463–471. DOI: 10.1016/J.RESS.2015.05.001.

Shi, Y. et al. (2020). "Multi-dimensional Lévy processes with Lévy copulas for multiple dependent degradation processes in lifetime analysis." In: *Quality Engineering* 32.3, pp. 434–448. DOI: 10.1080/08982112.2020.1757704.

Shin, J. H. and Jun, H. B. (2015). "On condition based maintenance policy." In: *Journal of Computational Design and Engineering* 2.2, pp. 119–127. DOI: 10.1016/j.jcde.2014.12.006.

Silver, D. et al. (2016). "Mastering the game of Go with deep neural networks and tree search." In: *Nature* 529.7587, pp. 484–489. DOI: 10.1038/nature16961.

Sun, Q., Ye, Z.-S., and Chen, N. (2018). "Optimal Inspection and Replacement Policies for Multi-Unit Systems Subject to Degradation." In: *IEEE Transactions on Reliability* 67.1, pp. 401–413. DOI: 10.1109/TR.2017.2778283.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. 2nd ed. MIT press.

Tabikh, M. (2014). "Downtime cost and Reduction analysis: Survey results." PhD thesis. Mälerdalen University.

Thomas, D. S. (2018). *The Costs and Benefits of Advanced Maintenance in Manufacturing*. Tech. rep. US Department of Commerce, National Institute of Standards and Technology. DOI: 10.6028/NIST.AMS.100-18.

Vachon, W. (2002). "Long-term O&M costs of wind turbines based on failure rates and repair costs." In: *WINDPOWER, American Wind Energy Association annual conference*. Portland, pp. 2–5.

van der Duyn Schouten, F. A. and Vanneste, S. G. (1990). "Analysis and computation of (n, N)-strategies for maintenance of a two-component system." In: *European Journal of Operational Research* 48.2, pp. 260–274. DOI: 10.1016/0377-2217(90)90379-P.

Veldman, J., Klingenberg, W., and Wortmann, H. (2011). "Managing condition-based maintenance technology: A multiple case study in the process industry." In: *Journal of Quality in Maintenance Engineering* 17.1, pp. 40–62. DOI: 10.1108/13552511111116240.

Vu, H. C. et al. (2014). "Maintenance grouping strategy for multi-component systems with dynamic contexts." In: *Reliability Engineering and System Safety* 132, pp. 233–249. DOI: 10.1016/j.ress.2014.08.002.

Waeyenbergh, G. and Pintelon, L. (2002). "A framework for maintenance concept development." In: *International Journal of Production Economics* 77.3, pp. 299–313. DOI: 10.1016/S0925-5273(01)00156-6.

Wang, H. (2002). "A survey of maintenance policies of deteriorating systems." In: *European Journal of Operational Research* 139.3, pp. 469–489. DOI: 10.1016/S0377-2217(01)00197-7.

Welte, T. M., Vatn, J., and Heggset, J. (2006). "Markov State Model for Optimization of Maintenance and Renewal of Hydro Power Components." In: *2006 International Conference on Probabilistic Methods Applied to Power Systems*. IEEE, pp. 1–7. DOI: 10.1109/PMAPS.2006.360311.

Xia, L., Zhao, Q., and Jia, Q.-S. (2008). "A Structure Property of Optimal Policies for Maintenance Problems With Safety-Critical Components." In: *IEEE Transactions on Automation Science and Engineering* 5.3, pp. 519–531. DOI: 10.1109/TASE.2007.910763.

Xu, J. et al. (2021). "Generalized condition-based maintenance optimization for multi-component systems considering stochastic dependency and imperfect maintenance." In: *Reliability Engineering & System Safety* 211, p. 107592. DOI: 10.1016/j.ress.2021.107592.

Zhang, N. et al. (2020). "Condition-based maintenance for a K-out-of-N deteriorating system under periodic inspection with failure dependence." In: *European Journal of Operational Research* 287, pp. 159–167. DOI: 10.1016/j.ejor.2020.04.041.

Zhang, Z. et al. (2018). "Degradation data analysis and remaining useful life estimation: A review on Wiener-process-based methods." In: *European Journal of Operational Research* 271.3, pp. 775–796. DOI: 10.1016/J.EJOR.2018.02.033.

Zhou, Y. et al. (2013). "Maintenance optimisation of a multi-state series–parallel system considering economic dependence and state-dependent inspection intervals." In: *Reliability Engineering & System Safety* 111, pp. 248–259. DOI: 10.1016/J.RESS.2012.10.006.

Zhu, Q., Peng, H., and Houtum, G. J. van (2015). "A condition-based maintenance policy for multi-component systems with a high maintenance setup cost." In: *OR Spectrum* 37.4, pp. 1007–1035. DOI: 10.1007/s00291-015-0405-z.

{2.7182818284}