Thesis submitted in partial fulfilment of the requirements of the degree Doctor rer. nat. of the Faculty of Mathematics and Natural Science of the Bergische Universität Wuppertal

Models and Methods for reliability based Maintenance Scheduling

Michael Gröger

November 2016

1st Supervisor: Prof. Dr. Hanno Gottschalk
2nd Supervisor: Prof. Dr. Kathrin Klamroth

Die Dissertation kann wie folgt zitiert werden:

urn:nbn:de:hbz:468-20170309-100548-6
[http://nbn-resolving.de/urn/resolver.pl?urn=urn%3Anbn%3Ade%3Ahbz%3A468-20170309-100548-6]

Contents

List of Figures

List of Tables

List of Algorithms

1. Introduction

Maintenance plays an important role in gas turbine operation, because it ensures the reliable and safe operation. Next to fuel consumption, service cost are the second largest portion of life cycle cost in the operation of a gas turbine. Therefore the outage schedule for a gas turbine is important.

Today, predictive maintenance actions for a gas turbine are determined by one global life counter $c_{\text{global}}(t)$ which represents the engine model with regards to the consumed life of the gas turbine, confer [57]. If the gas turbine reaches the life counter limit l_{g} for the respective maintenance interval, then the operator has to carry out the respective service according to its maintenance manual. This approach is based on deterministic lifetime limits. It does not utilize the full lifetime of the individual components. With certain changes in this set up, we can improve the flexibility and performance of a maintenance schedule.

The main change to overcome this drawback is to establish a multi life counter approach. With this idea, we track life consumption on parts or component level. Further, we will consider the probabilistic nature of failure mechanism which affects the gas turbine and take into account the risk connected to a failure. With these improvements the maintenance concepts become more flexible, but it is also a harder task to find a good tailor-made maintenance schedule.

There are many mathematical tools, which help us to quantify the risk of a failure. We have stochastic methods like reliability statistics to describe the risk of a failure, confer [34]. Further, we can use stochastic models to represent the probability of a failure for a certain failure mechanism which affects a component. For example in [80] a probabilistic low cycle fatigue (LCF) model was presented. Also methods from insurance mathematics help us to model the associated cash flows in our maintenance models, confer [56]. All of these methods give us the opportunity to model how the maintenance concept affects the key performance indicators (KPI) of the gas turbine operator, like for example availability, performance or revenue. In this thesis we will focus on revenue.

The mentioned methods will help us to model our maintenance problem for a fixed maintenance concept. Therefore, we use stochastic modeling and optimization methods to create optimization problems which improve the KPI's. This is the main focus of the thesis.

In this thesis we present three different modeling methods which use three different mathematical methodologies. We use an impulse control approach, confer [12] and [24], Dynamic Programming, confer [10], and a partially observable Markov Decision Process, confer [19]. With the help of these models, we are able to create optimization problems for our maintenance models. These methods give us the possibility to establish the optimal timing for the two service actions of type replacement and inspection which are commonly used service types, confer [65]. It is important to distinguish these two service actions. In general a replacement sets a component into an as new condition and we know the state of the desired component exactly. An extension of this model is that we allow the repair after a failure to a "as new" condition. In contrast, in the case of an inspection, we do not know the state of the component exactly. The inspection gives us information about the exact actual state of the component. The gathered information can lead to a premature replacement. This observation increases the complexity of models which include inspection. But in real applications, we often find the service action of type inspection. Further, we have to adjust, if possible, our probabilistic failure models such that they can make use of inspection. In general, an inspection can reduce the risk of a failure, because it readjusts the failure probability like in the case of failures due to cracks. But this is not possible for all failure mechanisms like for example LCF or high cycle fatigue (HCF). In [2] respectively in [83], it was proved that the Dynamic Programming and POMDP approaches can be used for maintenance modeling.

A further important part of this thesis is the numerical implementation of the mentioned methods. We show that it is possible to implement them for real sized problems and that we can find exact or approximate solutions within acceptable time limits. Today, many personal computers have multi processor or multi core systems. Therefore, we do also speed up the algorithm with the use of parallelization. This is often possible, because the methods often include loops and the single loop steps do not depend on each other.

In this thesis we use the well known approaches like mentioned in the paragraph before the last one. For the impulse control problem approach, we use the idea of the algorithm presented in [22] and we extend the introduced maintenance model in [23] through a more complex one with multiple components. As presented in [2], we use dynamic programming for maintenance scheduling. Therefore, we implement the basic backward algorithm from [10] and extend it to an algorithm which is suitable for a parallel implementation. Further, we use methods from [73] to build an approximate dynamic programming algorithm. We exploit the structure of our problem and use a parallel implementation. As last step with use the algorithm from [18] and [32] to solve our POMDP problem. To speed up the algorithms, we extend them so that we can use a multi processor environment. Despite many of the approaches and algorithms used in this thesis are well known, a systematic comparison and application to real world problems has been missing so far. In particular this applies to the combination of risk based maintenance scheduling with probabilistic life calculation. In this thesis it is shown for the first time, how the probabilistic characterization of material failure

can be included into the optimization of maintenance intervals. Further, we present for the first time a parallel implementation of some algorithms mentioned before.

As last step of the short introduction we outline the structure of the thesis shortly. In Chapter 2, we give a short overview about actual and future maintenance concepts. Therefore, we introduce how lifetime consumption can be tracked and calculated. For the future multiple life counter approaches we introduce a probabilistic LCF life prediction model based on the work of [80].

In Chapter 3 we present two simple toy models with one component respectively two components to introduce our basic idea of our maintenance modeling approach. We show for very idealized examples how we incorporate the notion of risk into our model and how we create an optimization problem from this idea.

In Chapter 4 we discuss our first "real" maintenance model which includes only the service type of replacement. More particularly, we set the component back into as new condition. We introduce an impulse control framework to solve the problem and further we establish basic maintenance modeling approaches. We follow the ideas from [12] for the impulse control theory and [22] for the solution method. We also present and discuss numerical results.

Chapter 5 deals with an advanced replacement model. The advanced model has the option to repair the gas turbine after a failure. This is an additional feature compared to the replacement model in Chapter 4. The solution strategy for the optimal service problem, posed by this model, is based on the mathematical theory of Dynamic Programming which we present briefly. We follow the ideas of [10]. Further, we establish approximate Dynamic Programming to reduce complexity in the Dynamic Programming framework like in [73]. This step is needed, because our approach suffers from the curse of dimensionality. At the end of the chapter we present numerical analysis for the runtime and compare the exact algorithm against the approximate algorithm.

In Chapter 6 we discuss our last model. First, we introduce the service action of inspection. In case of an inspection we replace parts only, if they do not fulfill certain criteria. To illustrate this, we establish a simple probabilistic crack growth model for inspection. Then we use this crack model in a partially observable Markov Decision Process to set up and solve our maintenance problem. For this model we also present two solution methods, one exact algorithm (Incremental-Pruning), confer [18], and one approximate algorithm (α-min-Algorithm), confer [32]. We present numerical results which compare the performance of both algorithm.

In the last Chapter a brief summary of the presented research work is provided and we give an outlook to future research based on the results achieved in this thesis.

In Appendix A, we present essential probability theory definitions and results.

2. Gas Turbine Maintenance Overview

A gas turbine is an internal combustion engine which consists of a compressor, a combustion chamber and a turbine. For a detailed cross sectional view, see Figure 2.1. The combustion chamber is located between the compressor and the turbine. The compressor takes the ambient air and compresses it during its way to the combustion chamber. In the combustion chamber the compressed air is mixed with fuel and ignited. The hot combustion gas flows into the turbine and expands there. During this process, the enthalpy of the high pressure fluid is transformed into mechanical energy. This energy is used to drive the compressor and an electrical generator. But some enthalpy is still remaining in the exhaust gas. This thermodynamical cycle is called Brayton cycle [57].

Based on this cycle, we can identify damage mechanisms which affect gas turbine lifetime and increase wearout. There can be erosion from dirt in the ambient air and the fuel. In combination with high temperatures this can lead to corrosion inside the gas turbine. Due to high thermal load in the combustion chamber and turbine section, there is creep damage and oxidation present. Creep is one of the main limiting life factors. The changing thermal and mechanical loads during operations lead to LCF (low cycle fatigue) damage. Further, components and parts are effected by HCF (High cycle fatigue) damage which occurs for example as a result of vibration at the blades, confer [57].

Based on the existence of the different damage mechanisms, the wearout of the gas turbine increases and the gas turbine becomes continuously more unreliable. The performance also decreases. The wearout can lead to failures of parts. The failures can cause performance loss or more dramatical consequences like unplanned outages, loss of revenue, total loss of the gas turbine or it affects human safety. The failure of a gas turbine reduces the reputation of the gas turbine manufacturer.

To reduce the mentioned risks, there is a need of maintenance for the gas turbine to negate them. Every gas turbine manufacturer sells its gas turbine with a maintenance concept. The concept consists of operating interval limits in terms of (equivalent) operating hours and starts and a list of corresponding service actions. Starts play an important role in the concepts, because they correspond to cyclic damage mechanisms like LCF or crack growth. The service scope includes a list of parts which are inspected or replaced. The interval limits prevent the operator to run the gas turbine parts over

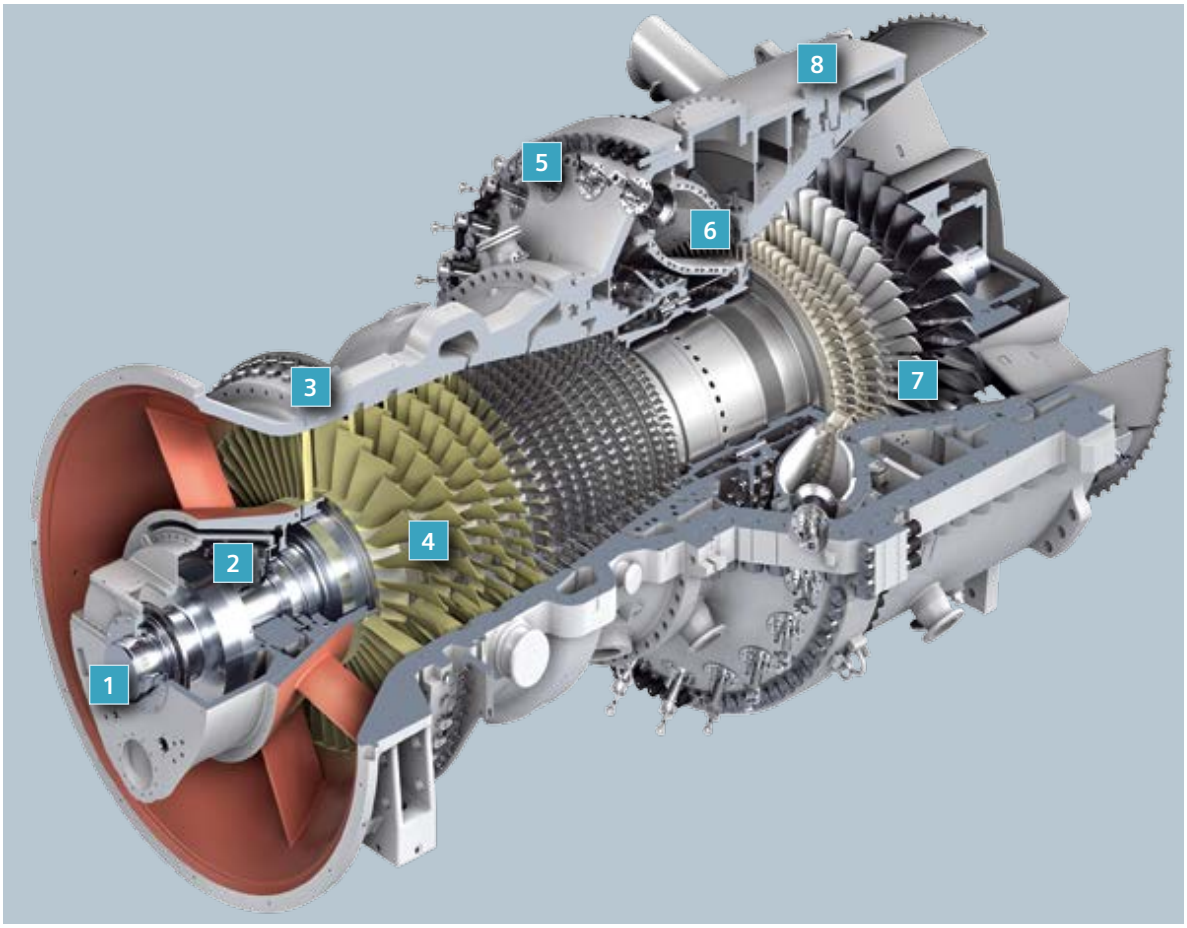


Figure 2.1.: 3d sectional view of a gas turbine: 1=Generator coupling; 2=Rotor bearings; 3=Variable inlet guide vanes; 4=Compressor; 5=Fuel nozzles, 6=Combustion chamber; 7=Turbine rotating blades; 8=Turbine casing. The figure is taken from [85].

their lifetime limits. Therefore, all risks connected to gas turbine operation within a tolerable limits.

It is important to predict the lifetime of components very accurately and to use this information to determine service interval lengths. Also, we need to count or track the life consumption of the gas turbine to schedule the outages and to stay below the lifetime limits.

This chapter contains a short overview of how lifetime limits are determined and how lifetime counting works. Further, it presents a summary about actual maintenance concepts and future maintenance concepts. The last section describes more closely the motivation for this thesis.

2.1. Lifetime Limits

We select LCF for polycrystalline metal as a model damage mechanism to show how we can determine the lifetime for a part. LCF for polycrystalline metal is important, because various expensive hot gas path (HGP) parts of the gas turbine are made of polycrystalline metal. Further, we introduce a probabilistic lifetime model for LCF, because our work is based on failure probabilities and not on deterministic lifetimes. We follow the main ideas from [35], [45] and [80] where a probabilistic LCF model was developed.

2.1.1. LCF Life Prediction

To describe the deformation of components made of polycrystalline metal under a cyclic load, we use linear isotropic thermoelasticity. This approach is based on continuum mechanics and assumes small deformations inside the component. We present a definition which gives us a partial differential equation to determine the displacement field of the component. It is taken from the work of [16] and [26]. It also establishes the important term of the stress tensor.

Definition 2.1 (Mixed Problem of Linear Isotropic Thermoelasticity)
Let $\Omega \subset \mathbb{R}^3$ be a domain with piecewise Lipschitz boundary, let ν be the normal of the boundary $\partial\Omega$ and let $f : \Omega \to \mathbb{R}^3$ be an external load. Let $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ be a partition and let $g : \partial\Omega_N \to \mathbb{R}^3$ be a surface load on $\partial\Omega_N$. Then, the mixed problem of linear isotropic thermoelasticity is given by the boundary value problem for the component displacement field $u : \Omega \to \mathbb{R}^3$:

$$\begin{aligned}
\nabla\sigma^e(u) + f &= 0 && \text{in } \Omega,\\
\sigma^e(u) &= \mu\left[\nabla u + \nabla u^T\right] + \left[\lambda(\nabla u) - \alpha_{TEC}(3\lambda + 2\mu)(T - T_0)\right] && \text{in } \Omega,\\
u &= 0 && \text{on } \partial\Omega_D,\\
\sigma^e(u)\cdot\nu &= g && \text{on } \partial\Omega_N,
\end{aligned}$$

where the temperature field T satisfies the boundary value problem

$$\Delta T = 0 \quad \text{in } \Omega$$

with either one of the following thermal boundary conditions

$$\begin{aligned}
T - T_0 &= T_s && \text{on } \partial\Omega && \text{(prescribed temperature)},\\
\kappa\nabla T\cdot\nu &= q && \text{on } \partial\Omega && \text{(prescribed heat flux)},\\
\kappa\nabla T\cdot\nu &= \alpha_{HTC}(T - T_e) && \text{on } \partial\Omega && \text{(heat transfer to ambient)}.
\end{aligned}$$

Here, T_0 is the reference temperature field at which the displacement field is everywhere zero and T_e is the temperature of the ambient at a distance far from $\partial\Omega$. λ and μ

are called Lame coefficients, α_{TEC} thermal extension coefficient, α_{HTC} heat transfer coefficient and κ heat conduction coefficient. q is the amount of heat flux into the boundary surface by an outside source. The components of the (elastic) stress tensor $\sigma^e(u) : \Omega \to \mathbb{R}^{3\times3}$ are denoted by $\sigma^e_{ij}(u)$, $i, j = 1, 2, 3$.

We can interpret the stress tensor σ^e as force per unit area which acts in a virtual cross section on an object. Another important quantity is the linearized strain rate tensor ϵ^e which is defined as

$$\epsilon^e : \Omega \to \mathbb{R}^{3\times3},\ \epsilon^e(u) = \frac{1}{2}\left(\nabla u + \nabla u^{\mathrm{T}}\right). \tag{2.1}$$

According to a generalization of Hooks law [90] this tensor depends on the stress tensor and describes the relative displacement of the material under external load.

2.1.2. Fatigue

In 1858 August Wöhler analyzed fatigue systematically, confer [97]. He realized that material can be much easier damaged under cyclic loading than under static load. This is very important, because in many engineering applications cyclic loading is present like in a start-stop process (start and shutdown of a gas turbine). We can use fatigue analysis to determine the number of cycles until a failure occurs. Failure means that the material properties are so much degenerated that the considered object can not fulfill certain design criteria. For polycrystalline metal, which is used for turbine blades, the fatigue process can be roughly divided in three parts:

1. **Crack initiation**
2. **Stable crack growth under cyclic loading**
3. **Residual fracture**

For more detailed explanation of the three phases confer for example [88]. In the next step, we derive a model to estimate the time n in terms of cycles to crack initiation following closely [76]. The model is based on tests with specimens which are exposed to cyclic load. The test determines the number of cycles N until failure under a given stress amplitude $\sigma_a = (\sigma_{\max} - \sigma_{\min})/2$. The results are presented in $S - N$ diagrams, confer Figure 2.2. These curves are called "Wöhler curves". Another approach for the test procedure is, to use a strain controlled test, where the strain amplitude $\epsilon_a = (\epsilon_{\max} - \epsilon_{\min})/2$ is given instead of a stress amplitude σ_a. The Coffin-Manson-Basquin (CMB) equation is a quantitative description of Wöhler curve for the strain case. The equation is given by

$$\epsilon_a = \frac{\sigma'_f}{E}(2N_i)^b + \epsilon'_f(2N_i)^c \tag{2.2}$$

with Young's modulus E, the fatigue strength coefficient σ'_f, the fatigue strength exponent b, fatigue ductility coefficient ϵ'_f and fatigue ductility exponent c. The equa-

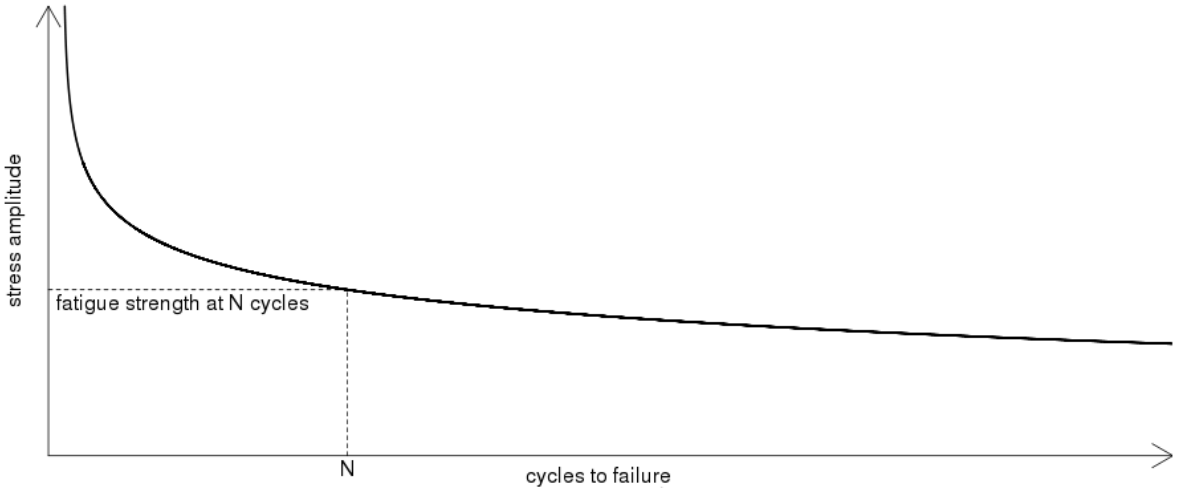


Figure 2.2.: $S - N$ diagram.

tion describes the relation between the number of cycles N_i until crack initiation and the strain amplitude ϵ_a. The first summand is called Basquin equation and can be used to describe the elastic part of the Wöhler curve. The second summand is called Coffin-Manson equation, which describes the plastic range of the Wöhler curve. The parameters can be estimated from test data.

With the help of the CMB equation we can predict the lifetime of a component. Therefore we search in the component for the location with the highest strain amplitude ϵ_a and calculate with the CMB equation the maximal number of cycles $N_{\max}$ until failure. The Number $N_{\max}$ gives us a predicted lifetime end. This gives us an important information for our maintenance concept.

In general we can not use the CMB equation directly. As we see in Figure 2.3, the CMB equation is a fit trough the data points and we see a large scatter in the data points which gives us an uncertainty. The origin of the uncertainty comes from the natural scatter in the material parameters. To cover the uncertainty, we have to shift the curve into a more conservative position. The method is called safe life approach or deterministic lifetime. Further, we include thermal dependency of the material parameters to take the material's thermal behavior into account.

Now, we want to introduce a local probabilistic model for LCF. The model takes size effects, inhomogeneous strain, temperature fields into account and the scatter in the material parameters. It gives us an alternative to the use of safety factors.

2.1.3. A probabilistic LCF Model

We consider a time continuous failure process like in [37] and [65]. Let N denote a continuous random variable on some probability space with range in $\mathbb{R}_+$. We call N

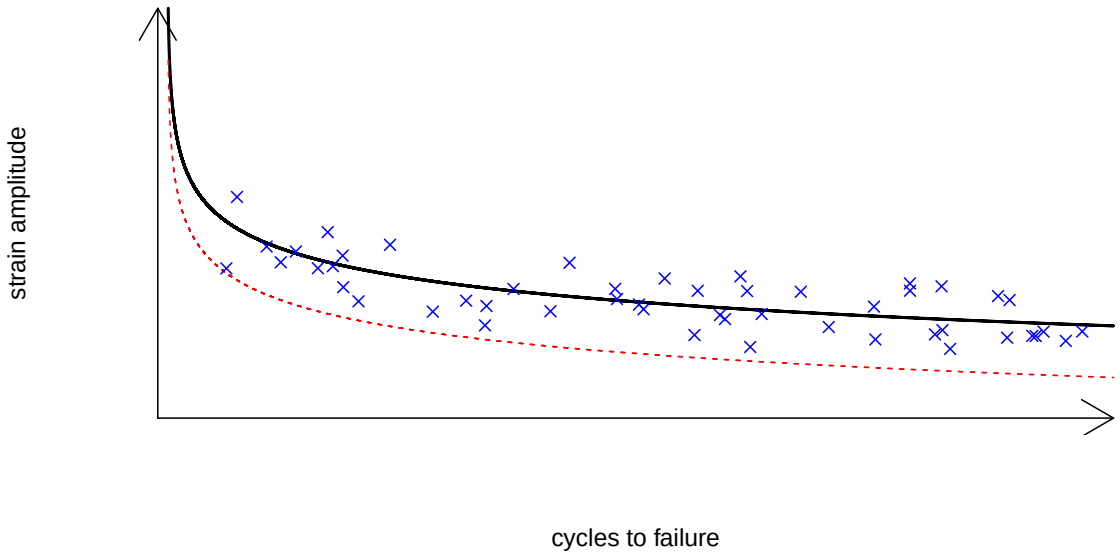


Figure 2.3.: $E-N$ diagram with safe life approach. The black line represents the fitted CMB equation and the dashed red line the transformed curve under the safe life approach.

failure time or number of cycles until failure. Let Pr be the underlying probability measure. Then

$$F(n) = Pr(n \leq N) \tag{2.3}$$

is the cumulative distribution function and

$$f(n) = \frac{d}{dn} F(n) \tag{2.4}$$

the density function. The hazard rate function or instantaneous failure rate function is denoted by

$$h(n) = \lim_{\Delta n \to 0} \frac{Pr(n < N < n + \Delta n | N > n)}{\Delta n} \tag{2.5}$$

and it will be important in our modeling approach. Here, $Pr(A|B)$ is the conditional probability, confer Appendix A. There also exists the cumulative hazard rate function

$$H(n) = \int_0^n h(\tau)\, d\tau. \tag{2.6}$$

There are two relationships between the previously defined functions in equations (2.3) to (2.6) given by

$$h(n) = \frac{f(n)}{1 - F(n)} \quad \text{and} \quad S(n) = 1 - F(n) = \exp(-H(n)). \tag{2.7}$$

$S(n)$ is called survival function. For a small "cycle/time step" Δn, the product $h(n) \cdot \Delta n$ is an approximation for the conditional probability that there will be a failure in the next step Δn, if there was no one before n.

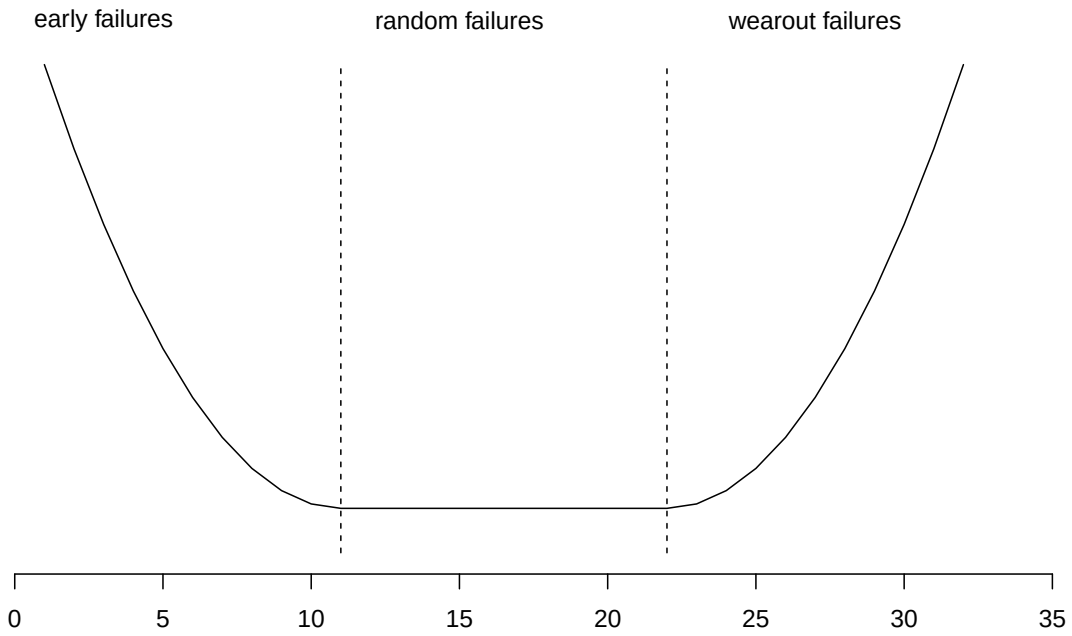


Figure 2.4.: Bathtub curve: First zone describes early failures, second zone random failures, third zone wearout failures.

For our model we need an important property of the hazard rate. Let N_1 and N_2 be two independent continuous random variables with range in $\mathbb{R}_+ := \{x \in \mathbb{R} \colon x \geq 0\}$ and their corresponding hazard rates h_1 and h_2. Then the random variable $N = \min(N_1, N_2)$ has the hazard rate $h = h_1 + h_2$. This corresponds to a serial connection of different systems compared to Section 4.3.2. Each of the four previously defined functions, from Equations (2.3) to (2.6), can be used as a characterization of our probabilistic failure model. The hazard rate function is an often used approach to model failure times in reliability engineering. We show the concept of hazard rate modeling in Figure 2.4. This curve is called bathtub curve and is divided into three areas.

The early failure period ("infant mortality") with a falling hazard rate which refers to failures due to assembling or material faults. The period within the normal lifetime has a constant hazard rate that describes the occurrence of random failures. The final segment with increasing hazard rate that originate from wearout at the end of the regular lifetime. In this work we consider only the last part of the bathtub curve and neglect the other parts for simplicity.

After this introduction, we can start with the description of the model following [45] and [80]. We start with a geometry $\Omega \subset \mathbb{R}^3$ of a component and consider a LCF failure process on the boundary $\partial\Omega$. We also consider a random failure time N in terms of cycles and we subdivide the surface of Ω into a partition $\{A_i\}_{i=1,\dots,m}$ of $\partial\Omega$, e.g.

$$\partial\Omega = \bigcup_{i=1}^{m} A_i \quad \text{and} \quad A_i \cap A_j = \emptyset \quad \text{for } i \neq j.$$

We present two important assumptions for our LCF model which are established in [45]:

Assumption 2.2
The LCF failure process on $\partial\Omega$ induces a failure process on each A_i with crack initiation times N_i, $i = 1, \ldots, m$, such that the random variables N_i, $i = 1, \ldots, m$, are independent.

If the assumption holds, then we get $N = \min(N_1, \ldots, N_m)$ for the first LCF crack initiation time on $\partial\Omega$ and we can represent the hazard rate h of N according to Section 4.3.2 as

$$h = \sum_{i=1}^{m} h_i$$

with $h_i = h_{A_i}$ is the hazard rate according to N_i, $i = 1, \ldots, m$ respectively connected to the surface region A_i. For polycrystalline metal this assumption is reasonable in the phase prior to significant crack growth and we introduce the stronger assumption:

Assumption 2.3
In any measurable surface region $A \subseteq \Omega$, the corresponding hazard rate h_A is a local functional of the displacement field u and of the temperature field T in that particular region:

$$h_A(n) = \int_A \rho\left(n; \nabla u, \nabla^2 u, T\right) dA.$$

We restrict ourselves to the second assumption and change the dependency of ρ from u to the strain amplitude field ϵ_a on $\partial\Omega$ and we get

$$h(n) = \int_{\partial\Omega} \rho(n; \epsilon_a, T)\, dA.$$

With this approach and the relation $F(n) = 1 - \exp(-H(n))$, we obtain for the probability for LCF crack initiation on $\partial\Omega$ until cycle n the following expression:

$$\begin{aligned} F(n) &= 1 - \exp(-H(n)) \\ &= 1 - \exp\left(-\int_0^n h(\tau)\, d\tau\right) \\ &= 1 - \exp\left(-\int_0^n \int_{\partial\Omega} \rho(n; \epsilon_a, T)\, dA d\tau\right). \end{aligned} \tag{2.8}$$

As last step we need an explicit representation for hazard density ρ in our approach. We use a Weibull model,

$$F(n) = 1 - \exp\left(-\left(\frac{n}{\eta}\right)^m\right) \quad \text{and} \quad h(n) = \frac{m}{\eta}\left(\frac{n}{\eta}\right)^{m-1},$$

to model the time N to crack initiation. We use the following hazard function

$$\rho(n; x) = \rho(n; \epsilon_a(x), T(x)) = \frac{m}{N_{i_{\text{det}}}(\epsilon_a(x), T(x))}\left(\frac{n}{N_{i_{\text{det}}}(\epsilon_a(x), T(x))}\right)^{m-1}. \tag{2.9}$$

The local scale parameter $N_{i_{\text{det}}} = N_{i_{\text{det}}}(\epsilon_a(x), T(x))$ gives us a link to the deterministic CMB equation with an underlying temperature model. We use $m \in (0, \infty)$ as shape parameter of the Weibull model to model the scatter. A large value of m gives us a small scatter and $m = \infty$ is the deterministic lifetime limit. The choice $m \geq 1$ is suitable for a LCF failure mechanism, because the hazard rate increases over the time.

To get our final cumulative failure distribution function for LCF failure, we combine equation (2.8) and equation (2.9) and we obtain

$$F(n) = 1 - \exp\left(-\int_0^n \int_{\partial\Omega} \frac{m}{N_{i_{\text{det}}}(\epsilon_a(x), T(x))} \left(\frac{\tau}{N_{i_{\text{det}}}(\epsilon_a(x), T(x))}\right)^{m-1} d\tau dA\right) \tag{2.10}$$

$$= 1 - \exp\left(-\int_{\partial\Omega} \left(\frac{n}{N_{i_{\text{det}}}(\epsilon_a(x), T(x))}\right)^{m} dA\right). \tag{2.11}$$

We note that in Equation (2.11) the expression $N_{i_{\text{det}}}(\epsilon_a(x), T(x))$ has the function of scale parameter η of the Weibull model and $N_{i_{\text{det}}}$ has the units $[N_{i_{\text{det}}}] = \text{cycles} \times \text{meter}^2$. According to our derivation, we can define our model as presented in [80]:

Definition 2.4 (Local probabilistic LCF Model)
Let $\Omega \subset \mathbb{R}^3$ be a domain with boundary $\partial\Omega$ representing a geometry which is exposed to a cycling load. Further let the scale field $N_{i_{det}} = N_{i_{det}}(\epsilon_a(x),)$, $x \in \partial\Omega$, be the solution of the CMB equation

$$\epsilon_a(x) = \frac{\sigma_f'(T(x))}{E(T(x))} (2N_{i_{det}}(x))^{b(T(x))} + \epsilon_f'(T(x)) (2N_{i_{det}}(x))^{c(T(x))}, \ x \in \partial\Omega, \tag{2.12}$$

where ϵ_a is the strain amplitude field and T the temperature field which determines the CMB parameters via a suitable temperature model. Then the local probabilistic LCF model is given by the cumulative distribution function in equation (2.10) for $n \in \mathbb{R}_+$ and $m \geq 1$, which yields the probability for LCF crack initiation in the interval $[0, n]$.

There are different ways to estimate the missing parameters in our model. We can determine the temperature and strain amplitude fields by experiments or simulate it by a finite element analysis (FEA), like in [81], or use fleet experience. We also have to calibrate CMB-parameters by experiments, confer [82].

To determine the lifetime in a probabilistic model we can choose an acceptable risk using equation (2.10) and the following equation

$$N_{\max}(F_{\text{limit}}) = \max\{n : F(n) \leq F_{\text{limit}}\}$$

to determine the maximal number $N_{\max}(F_{\text{limit}})$ of tolerable cycles. This thesis will present better methods to calculate the number $N_{\max}$ of maximal cycles in a probabilistic context.

The general advantage of probabilistic lifetime models is that we get failure probabilities which help us to quantify the risk of a failure or the risk of an interval extension. Further, it helps to explain, how inspection influences our component lifetimes and outage scheduling, which we will explain in Section 6.2.1 in more detail.

2.2. Life Counter

With the methods of the last Section 2.1, we can determine lifetime limits of the gas turbine components and define service interval lengths. We must track the life consumption to stay in our limits and to have the opportunity to schedule the outages. For this reason, we introduce lifetime counters which are based on linear damage accumulation introduced by Miner [62] and Palmgren [67]. It is a well known concept for gas turbine's life consumption tracking, confer [57].

The idea behind counters is very simple. We assume that we have a lifetime limit n_1 for a reference start and stop cycle C_1 and a lifetime limit n_2 for a second start and stop cycle C_2. Then our lifetime end is reached if

$$m_1 \cdot \frac{1}{n_1} + m_2 \cdot \frac{1}{n_2} = \sum_{i=1}^{2} \frac{m_i}{n_i} = 1$$

holds. Here m_1 and m_2 are the numbers of cycles of type C_1 and C_2. To make the equation more clear and user friendly, we scale the last equation with the reference cycle lifetime limit and get

$$1 \cdot m_1 + m_2 \cdot \frac{n_1}{n_2} = \sum_{i=1}^{2} m_i \cdot \frac{n_1}{n_i} = n_1.$$

Now, every cycle of type C_1 is weighted with the factor 1 and the cycles of type C_2 are weighted with the factor $\frac{n_1}{n_2}$. We compare every portion of consumed life to a reference condition. This is the reason, why we talk about equivalent hours, starts or cycles. In a more elaborated method, we count the cycles with a more sophisticated method as for example the rainflow counting algorithm presented in [27].

We introduce two basic models for total lifetime counting, which are used for gas turbine life counting today, confer [25] and [57]:

- **EOH-Model:** The Equivalent Operating Hours-Model assumes that there is a connection between cyclic and time based damage. In this case a cyclic event reduces the number of allowable operating hours. Our counter is in general defined

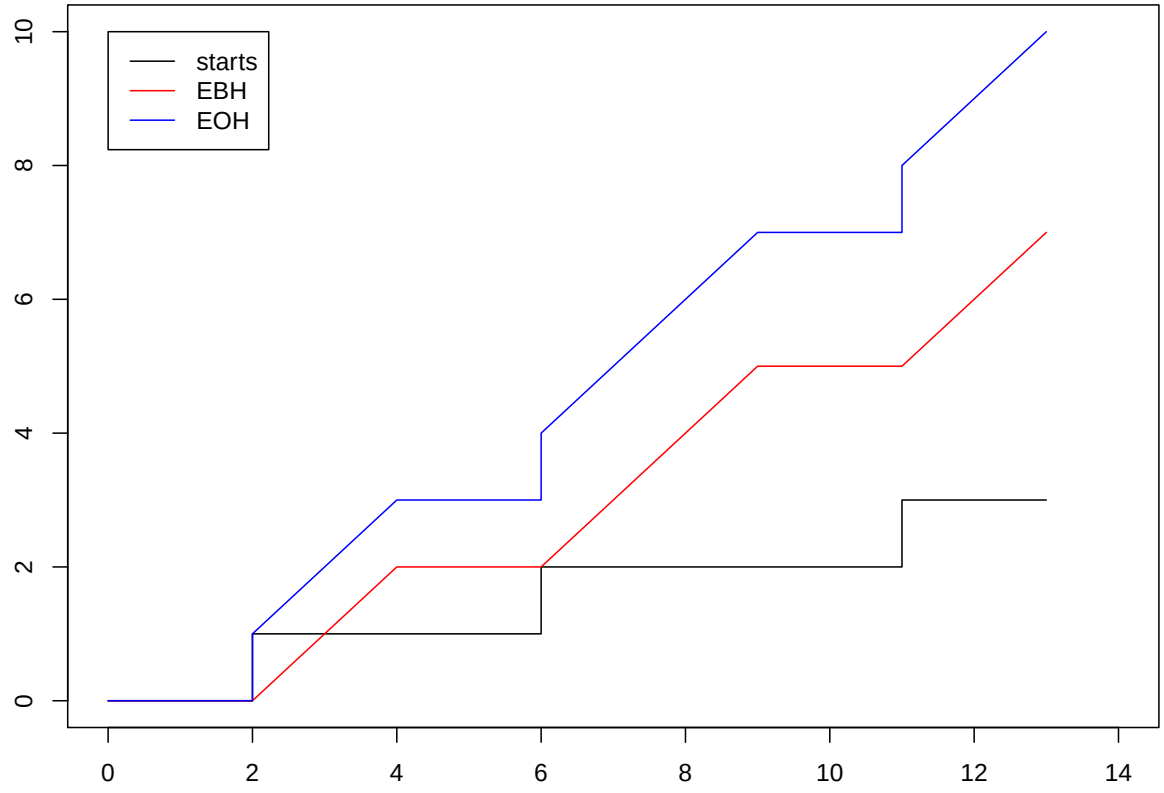


Figure 2.5.: Example ES, EBH, EOH counter trends for a fixed operating regime.

by two equations

$$t_{\mathrm{EOH}} = \sum_{i=1}^{\tilde{N}_{\mathrm{S}}} F_{\mathrm{S}i}\Delta_{\mathrm{S}i} + \sum_{i=1}^{N_{\mathrm{OH}}} F_{\mathrm{OH}i}\Delta_{\mathrm{OH}i}$$

$$n_{\mathrm{starts}} = \sum_{i=1}^{N_{\mathrm{s}}} 1$$

with N_{s} starts, start factor F_{S}, N_{OH} operating modes, operating factor F_{OH}, $\tilde{N}_{\mathrm{S}}$ different start types, $\Delta_{\mathrm{S}i}$ starts of type i and time $\Delta_{\mathrm{OH}i}$ in operating mode i. The factors F_{S} and F_{OH} represent the weights for the lifetime consumption in different operating modes. These factors depend on many input parameters like temperatures or load levels of the gas turbine. Starts are counted unweighted.

- **Box-Model:** The Box-Model assumes that there is no connection between cyclic and time based damage. In this case, we will count equivalent starts (ES) and

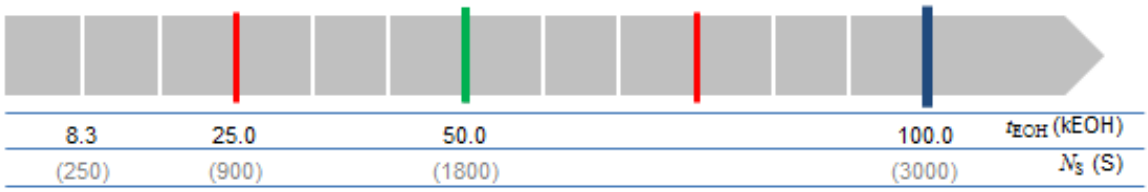


Figure 2.6.: EOH Arrow for maintenance concept representation. The white, red, green and dark blue vertical lines refer to different outage types. Every outage type is connected to a EOH and start limit.

equivalent baseload hours (EBH).

$$t_{\mathrm{EBH}} = \sum_{i=1}^{N_{\mathrm{OH}}} F_{\mathrm{OH}i}\Delta_{\mathrm{OH}i}$$

$$n_{\mathrm{ES}} = \sum_{i=1}^{N_{\mathrm{s}}} F_{\mathrm{S}i}\Delta_{\mathrm{S}i}$$

with N_{OH} different operating modes, operating factor F_{OH}, time $\Delta_{\mathrm{OH}i}$ in operating mode i, N_{s} different starts types, the start factor F_{S} and starts$\Delta_{\mathrm{S}i}$ for start mode i. The factors F_{S} and F_{OH} represent the weighting for the life consumption in different operating modes. These factors again can be weights depending on many input parameters like temperatures or load level of the gas turbine.

The factors in the equation consider the different operating modes of the gas turbine. In Figure 2.5 we see different examples of counters plotted over the time. We see that the ES counter has got the shape of a step function and increases with every GT start or cyclic event. The EBH counter is a continuous and monotone increasing function. The EOH counter is a mixture between the last two types. At every start there is a jump and after the start the counter will increase piecewise linearly in time.

2.3. Maintenance Concepts

We now have all tools, lifetime determination and lifetime counter, to create a maintenance concept. In general, a maintenance concept is a list of EOH or EBH/(E)S limits and a corresponding list of measures, like inspection or replacement of parts. The EOH or EBH (E)S limits represent the lifetime limits of the gas turbine parts.

For example, we illustrate a maintenance concept in Figure 2.6. There are four different outage types which are illustrated by the vertical lines in the gray arrow. Outage type 3 (green line) will be carried out every 50000 EOH, or 1800 starts, whichever occurs first. The list of measures is different for every outage.

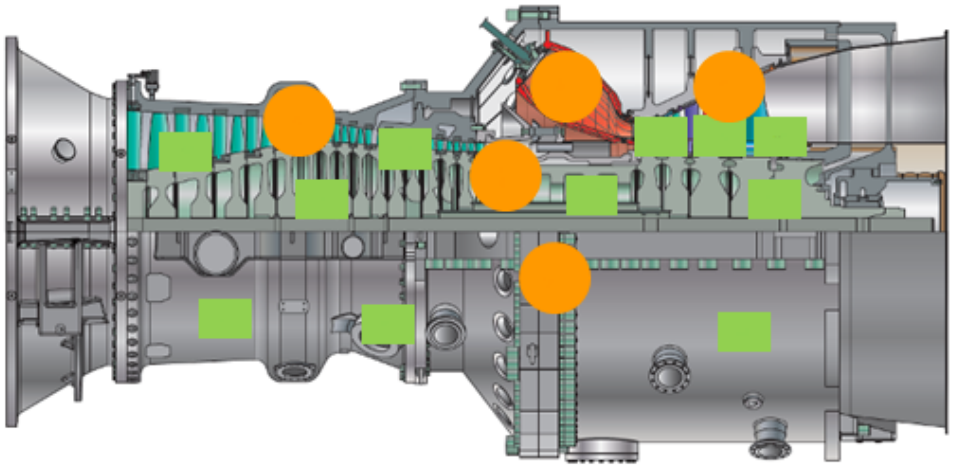


Figure 2.7.: A gas turbine overview with exemplary attached counters. The orange circles highlight part groups which need an exclusive life counter and the green boxes mark single parts.

It is very easy to determine the calendar date of an outage, because we only have to take into account, when the counter reaches a limit. Through the expected operating regime we can estimate the date.

2.3.1. Actual Situation and Outlook

Today, many gas turbine maintenance concepts are based on one lifetime counter of type EOH or BOX for the whole gas turbine. The counters include the different operating modes and regimes by factors in the counter equations. The limits represent the lifetime limits of the parts. Every outage type includes the replacement or inspection of different parts according to their individual lifetime limit.

With this concept, it is very easy to schedule the calendar dates of outages. We only have to consider the expected operating regime and predict the date of reaching the limit. If the operating regime of the gas turbine is constant or has got a periodic pattern, the outage plan will end up in a periodic pattern. This is very nice for planning outages. But there are some drawbacks. For example, if the outage is scheduled for a time with a high electrical price, then the operator has two options. The operator could execute the outage and loose money or the operator could prepone the outage and loose remaining lifetime of the gas turbine parts. Waste of remaining lifetime of parts is also lost money. In the end there is always some financial disadvantage. A second drawback is the existence of only one counter for the whole gas turbine. As a result of this, there will be often a waste of lifetime for specific parts. For example, if the gas turbine uses contaminated fuel, then the operating factor $F_{\mathrm{OH}i}$ of the counter

will increase. The accelerated counting process is a result of faster wearout of the hot gas path parts due to the fuel. Therefore, the replacement and inspection of compressor parts will be preponed while the wearout of the compressor is not influenced by the fuel.

Due to the missing flexibility of the one counter concept and the deterministic lifetime limits, there is the need to develop new maintenance concepts. One new idea is to use multiple lifetime counters on part level and to consider failure probabilities instead of deterministic lifetimes. Every important part or part group gets its own lifetime counter to consider the parts specific lifetime consumption. In Figure 2.7 relevant parts or components are marked which are candidates for individual lifetime counter. Here, important means that a part is very expensive or the overall reliability / availability has got a strong dependency of this part. The failure probability gives us more flexibility in the interval lengths, if we take the involved risks into account. In this case we are able to relax the interval limits, if the economic reward is promising.

The multiple counter concepts establish a new problem. It is now more difficult to create a good outage schedule, because we have more lifetime counters and soft limits which determine the outages. The simplest solution is to schedule an outage, if a counter reaches its deterministic lifetime limit. This approach is not the best, because there is benefit to combine outages for different parts. The advantage is available, because there are certain dependencies in the dismantling process of a gas turbine which will save time and increase the availability of the gas turbine or we can save cost connected to the disassemble process. Through good outage scheduling, we can reduce service cost of the gas turbine or increase availability. In the end, the gas turbine will become more profitable and reliable. Figure 2.8 illustrates the multiple counter problem.

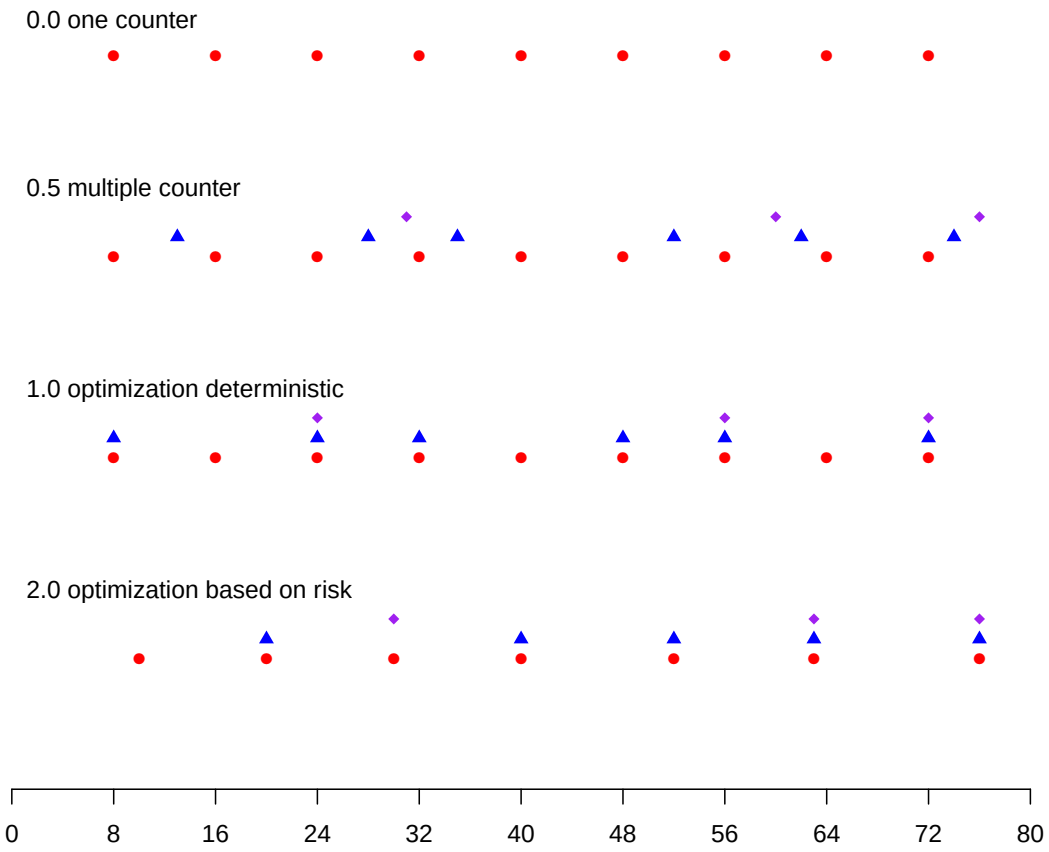


Figure 2.8.: Outage schedule problem: 0.0 one counter schedule; 0.5 multiple counter and start every time a outage if a limit is reached; 1.0 multiple counter and deterministic lifetime limits with optimization; 2.0 multiple counter and failure probabilities with optimization. The x-axis represents the time in kh.

3. The Toy Model

In this chapter, we present a method to calculate an "optimal" outage schedule for a simple toy model. It represents the basic idea for our maintenance scheduling approach. We try to maximize expected revenue for the gas turbine operator. Therefore, we analyze the operator's cash flow and maximize it with the help of service actions. The presented methods in this chapter and thesis work for every technical device. For the sake of simplicity we keep calling it gas turbine. This wording fits better in our context.

We consider a gas turbine which has only one lifetime counter and there is only one outage type. Further, the counter counts only real operating hours, because all factors in the counter equations are set to one. The outage sets the gas turbine in an as "new state" by replacing all critical parts. The gas turbine operator participates in the market of electricity where the operator sells electrical power for a constant price. The operator can order service at every time for a constant prize. Further, the operator has to pay a penalty fee, if the gas turbine fails. The gas turbine will not be replaced, if it fails. The operating regime of the gas turbine is constant and due to this fact the outage will be done in a periodic way. See Figure 3.1(a). In this case a good outage scheduling means to find an optimal interval length between two outages that maximizes the operator's profit.

We start with basic notations to model the problem as an optimization problem:

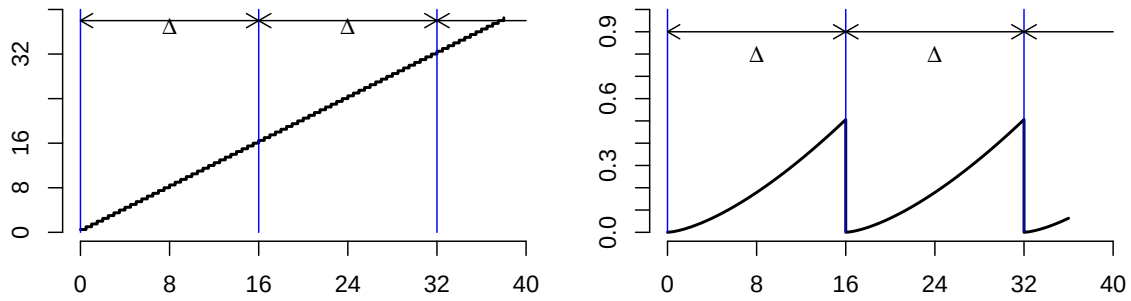
Definition 3.1 (Basic Definitions)
We define for our model the following parameters:

- *Time span between two service actions:* $\Delta > 0$.
- *Duration of the maintenance action:* $W > 0$.
- *Hazard rate (risk) of the gas turbine without maintenance:* $\hat{h}(t)$.
- *Cumulative hazard rate of the gas turbine without maintenance:*

$$\hat{H}(t) := \int_0^t \hat{h}(\tau)\, d\tau.$$

- *Survival function of the gas turbine without maintenance:*

$$\hat{S}(t) := \mathrm{e}^{-\hat{H}(t)} = \mathrm{e}^{-\int_0^t \hat{h}(\tau) d\tau}.$$



(a) Operating Regime (b) Hazard rate trend

Figure 3.1.: Schematic model representation of the toy model.

- *The operators revenue I, the financial risk of a gas turbine failure C_R, the maintenance cost C_M, nominal discount factor i_{no}.*
- *i-th operating interval:*

$$\mathbb{I}_i := [i(\Delta + W), (i+1)\Delta + iW], \ i = 1, \ldots.$$

- *The complete operating time:*

$$\mathbb{I} := \bigcup_{i=0}^{\infty} \mathbb{I}_i.$$

3.1. Modeling the one Component Case

Now, we deduce the maintenance model. Our first assumption is that after every service action the hazard rate / risk will be set to zero, i.e,

$$h(i(\Delta + W)) = 0, \ i = 0, 1, 2, \ldots$$

This idea is illustrated in Figure 3.1(b). We define the hazard rate of a gas turbine with periodic maintenance as

$$h(t) := \hat{h}(t \mod i(\Delta + W)) \text{ for } t \in \mathbb{I}_i, \ i = 0, 1, 2, \ldots. \tag{3.1}$$

We get the cumulative hazard rate $H(t)$ and the survival function $S(t)$ of the gas turbine with periodic maintenance directly by:

$$\begin{aligned} H(t) :&= \int_0^t h(\tau)\, d\tau = \int_0^t \hat{h}(\tau \mod \Delta + W)\, d\tau \\ &= \left\lfloor \frac{t}{\Delta + W} \right\rfloor \int_0^{\Delta} \hat{h}(\tau)\, d\tau + \int_0^{t \mod \Delta + W} \hat{h}(\tau)\, d\tau \\ &= \left\lfloor \frac{t}{\Delta + W} \right\rfloor \hat{H}(\Delta) + \hat{H}(t \mod \Delta + W) \end{aligned}$$

and

$$\begin{aligned} S(t) :&= \exp(-H(t)) = \exp\left(-\left\lfloor \frac{t}{\Delta+W} \right\rfloor \hat{H}(\Delta) - \hat{H}(t \mod \Delta+W)\right) \\ &= \exp\left(-\left\lfloor \frac{t}{\Delta+W} \right\rfloor \hat{H}(\Delta)\right) \exp\left(-\hat{H}(t \mod \Delta+W)\right) \\ &= \hat{S}(\Delta)^{\left\lfloor \frac{t}{\Delta+W} \right\rfloor} \hat{S}(t \mod \Delta+W), \end{aligned}$$

where $\lfloor \cdot \rfloor$ denotes the floor operator and it is defined by

$$\lfloor x \rfloor := \max\{y \in \mathbb{Z} \colon y \leq x\}.$$

With this informations we can model the cash flow of the gas turbine operator. The operator earns a revenue I per unit of time, if the gas turbine is in operation. The operator has to pay C_{M}, if the gas turbine starts a maintenance action. The operator loses C_{R} units of money, if the gas turbine fails. We will use the present value formulation with interest rate $i_{\mathrm{no}} > 0$. In summary we get the operators discounted cash flow by

$$\begin{aligned} \mathrm{pv}(\tau) &= \mathrm{e}^{-i_{\mathrm{no}}\tau} \cdot I \cdot \chi_{\{\text{gas turbine in operation and not failed until } \tau\}} \\ &\quad - \mathrm{e}^{-i_{\mathrm{no}}\tau} \cdot C_{\mathrm{R}} \cdot \delta_{\{\tau \text{ is equal to failure time}\}} \\ &\quad - \mathrm{e}^{-i_{\mathrm{no}}\tau} \cdot C_{\mathrm{M}} \cdot \delta_{\{\tau \text{ equal outage start}\}} \chi_{\{\text{gas turbine did not fail until } \tau\}} \end{aligned} \tag{3.2}$$

with the indicator function

$$\chi_{\{f(t)\}} := \begin{cases} 1, & f(t) \text{ is true} \\ 0, & \text{else} \end{cases}$$

and the delta distribution

$$\delta_{\{t\}} := \begin{cases} +\infty, & t = 0 \\ 0, & \text{else} \end{cases} \quad \text{with} \quad \int_{-\infty}^{+\infty} \delta(t - t_0) f(t)\, dt = f(t_0).$$

In the next step we introduce the random failure time T_{f} and we rewrite the cash flow equation equation (3.2)

$$\begin{aligned} \mathrm{pv}(\tau) &= \mathrm{e}^{-i_{\mathrm{no}}\tau} \cdot I \cdot \chi_{\{\tau < T_{\mathrm{f}}\}} \cdot \chi_{\{\tau \in \mathbb{I}\}} \\ &\quad - \mathrm{e}^{-i_{\mathrm{no}}\tau} \cdot C_{\mathrm{R}} \cdot \delta_{\{\tau - T_{\mathrm{f}}\}} \\ &\quad - \mathrm{e}^{-i_{\mathrm{no}}\tau} \cdot C_{\mathrm{M}} \cdot \delta_{\{\tau \bmod i(\Delta+W)\}} \cdot \chi_{\{\tau < T_{\mathrm{f}}\}}. \end{aligned}$$

The complete income of the operator in the interval $[0, t]$ is given by

$$\begin{aligned} \mathrm{PV}(t) &= \int_0^t \mathrm{pv}(\tau)\, d\tau \\ &= \int_0^t \mathrm{e}^{-i_{\mathrm{no}}\tau} \chi_{\{\tau < T_{\mathrm{f}}\}} \left(I \cdot \chi_{\{\tau \in \mathbb{I}\}} - C_{\mathrm{R}} \cdot \delta_{\{\tau - T_{\mathrm{f}}\}} - C_{\mathrm{M}} \delta_{\{\tau \bmod i(\Delta+W)\}}\right) d\tau. \end{aligned}$$

Since the lifetime T_f of the gas turbine is a random variable, we calculate the expected present value. The gas turbine operator earns only money at time t, if the gas turbine survives until t with the (survival) probability $S(t)$. The operator has to pay only for service action at t, if the gas turbine survives until t. But there is also a probability $S(t)\,h(t)\,d\tau$ that the gas turbine fails in the next time step $d\tau$, if the gas turbine survived until t. With this considerations we get the expected present value by

$$\mathrm{EPV} = \int_0^\infty \mathrm{e}^{-i_\mathrm{no}\tau}\, S(\tau) \left(I \cdot \chi_{\{\tau\in\mathbb{I}\}} - C_\mathrm{R} \cdot h(\tau) - C_\mathrm{M} \cdot \delta_{\{\tau \bmod i(\Delta+W)\}}\right) d\tau.$$

With the definitions from above we can eliminate the indicator functions and we obtain

$$\begin{aligned}
\mathrm{EPV} &= \int_0^\infty \mathrm{e}^{-i_\mathrm{no}\tau}\, S(\tau) \left(I \cdot \chi_{\{\tau\in\mathbb{I}\}} - C_\mathrm{R} h(\tau) - C_\mathrm{M} \delta_{\{\tau \bmod i(\Delta+W)\}}\right) d\tau \\
&= \underbrace{\sum_{i=0}^{\infty} \int_{i(\Delta+W)}^{(i+1)\Delta+iW} \mathrm{e}^{-i_\mathrm{no}\tau}\, S(\tau)\,(I - C_\mathrm{R} h(\tau))\, d\tau}_{\text{(a)}} \\
&\quad \underbrace{- \sum_{i=0}^{\infty} C_\mathrm{M}\, \mathrm{e}^{-i_\mathrm{no}((i+1)\Delta+iW)}\, S((i+1)\Delta + iW)}_{\text{(b)}}. \qquad (3.3)
\end{aligned}$$

Next, we simplify the first sum of (3.3):

$$\begin{aligned}
(3.3.\mathrm{a}) &= \sum_{i=0}^{\infty} \int_{i(\Delta+W)}^{(i+1)\Delta+iW} \mathrm{e}^{-i_\mathrm{no}\tau}\, S(\tau)\,(I - C_\mathrm{R} h(t))\, d\tau \\
&= \sum_{i=0}^{\infty} \int_0^{\Delta} \mathrm{e}^{-i_\mathrm{no}(\tau+i(\Delta+W))}\, S(\tau + i(\Delta+W))\,(I - C_\mathrm{R} h(\tau + i(\Delta+W)))\, d\tau \\
&= \sum_{i=0}^{\infty} \mathrm{e}^{-i_\mathrm{no} i(\Delta+W)} \int_0^{\Delta} \mathrm{e}^{-i_\mathrm{no}\tau}\, \hat{S}(\tau)\, \hat{S}(\Delta)^i \left(I - C_\mathrm{R} \hat{h}(\tau)\right) d\tau \\
&= \sum_{i=0}^{\infty} \mathrm{e}^{-i_\mathrm{no} i(\Delta+W)}\, \hat{S}(\Delta)^i \int_0^{\Delta} \mathrm{e}^{-i_\mathrm{no}\tau}\, \hat{S}(\tau) \left(I - C_\mathrm{R} \hat{h}(\tau)\right) d\tau.
\end{aligned}$$

Since

$$\mathcal{I}\left(\Delta, \hat{S}, \hat{h}, i_\mathrm{no}, I, C_\mathrm{R}\right) = \int_0^{\Delta} \mathrm{e}^{-i_\mathrm{no}\tau}\, \hat{S}(\tau) \left(I - C_\mathrm{R} \hat{h}(\tau)\right) d\tau$$

does not depend on i and

$$\frac{\mathrm{e}^{-i_\mathrm{no}(i+1)(\Delta+W)}\, \hat{S}(\Delta)^{i+1}}{\mathrm{e}^{-i_\mathrm{no} i(\Delta+W)}\, \hat{S}(\Delta)^i} = \mathrm{e}^{-i_\mathrm{no}(\Delta+W)}\, \hat{S}(\Delta) < 1$$

holds, the geometric series properties are true and we obtain

$$(3.3.\mathrm{a}) = \mathcal{I}\left(\Delta, \hat{S}, \hat{h}, i_\mathrm{no}, I, C_\mathrm{R}\right) \sum_{i=0}^{\infty} \mathrm{e}^{-i_\mathrm{no} i(\Delta+W)}\, \hat{S}(\Delta)^i = \frac{\mathcal{I}\left(\Delta, \hat{S}, \hat{h}, i_\mathrm{no}, I, C_\mathrm{R}\right)}{1 - \mathrm{e}^{-i_\mathrm{no}(\Delta+W)}\, \hat{S}(\Delta)}.$$

In the second sum of equation (3.3), we can use the same geometric series arguments as before and we obtain

$$\begin{aligned}(3.3.\text{b}) &= \sum_{i=0}^{\infty} C_{\text{M}}\, \mathrm{e}^{-i_{\text{no}}((i+1)\Delta+iW)}\, S\left((i+1)\,\Delta + iW\right) \\ &= \sum_{i=0}^{\infty} C_{\text{M}}\, \mathrm{e}^{-i_{\text{no}}((i+1)\Delta+iW)}\, \hat{S}\left(\Delta\right)^{i+1} \\ &= \frac{C_{\text{M}}\, \mathrm{e}^{-i_{\text{no}}\Delta}\, \hat{S}\left(\Delta\right)}{1 - \mathrm{e}^{-i_{\text{no}}(\Delta+W)}\, \hat{S}\left(\Delta\right)}.\end{aligned}$$

Finally, we get for the objective function the following representation

$$\begin{aligned}(3.3) &= \frac{\mathcal{I}\left(\Delta, \hat{S}, \hat{h}, i_{\text{no}}, I, C_{\text{R}}\right)}{1 - \mathrm{e}^{-i_{\text{no}}(\Delta+W)}\, \hat{S}\left(\Delta\right)} - \frac{C_{\text{M}}\, \mathrm{e}^{-i_{\text{no}}\Delta}\, \hat{S}\left(\Delta\right)}{1 - \mathrm{e}^{-i_{\text{no}}(\Delta+W)}\, \hat{S}\left(\Delta\right)} \\ &= \frac{\mathcal{I}\left(\Delta, \hat{S}, \hat{h}, i_{\text{no}}, I, C_{\text{R}}\right) - C_{\text{M}}\, \mathrm{e}^{-i_{\text{no}}\Delta}\, \hat{S}\left(\Delta\right)}{1 - \mathrm{e}^{-i_{\text{no}}(\Delta+W)}\, \hat{S}\left(\Delta\right)} \\ &=: J\left(\Delta\right).\end{aligned}$$

The function $J\left(\Delta\right)$ gives the expected profit of the gas turbine operator who uses maintenance intervals of the length Δ for an infinite time horizon. Now, we can formulate our optimization problem as

$$\max_{\Delta} \frac{\mathcal{I}\left(\Delta, \hat{S}, \hat{h}, i_{\text{no}}, I, C_{\text{R}}\right) - C_{\text{M}}\, \mathrm{e}^{-i_{\text{no}}\Delta}\, \hat{S}\left(\Delta\right)}{1 - \mathrm{e}^{-i_{\text{no}}(\Delta+W)}\, \hat{S}\left(\Delta\right)} = \max_{\Delta} J\left(\Delta\right)$$

subject to

$$\Delta \geq 0.$$

3.2. Existence of a Solution

In this section we present the existence of a solution $\tilde{\Delta}$ of $\max_{\Delta \geq 0} J\left(\Delta\right)$, i.e.

$$\tilde{\Delta} = \arg\max_{\Delta \geq 0} J\left(\Delta\right).$$

We assume that $\hat{h}$, $\hat{S} \in C^1\left([0, \infty)\right)$ and that they satisfy the following properties

$$\hat{h}\left(t\right) \geq 0, \;\; \hat{h}'\left(t\right) > 0 \quad \text{for all } t \geq 0.$$

The mentioned assumptions are not too strong, because the risk of a failure increases over time and we can claim $\hat{h}'\left(t\right) > 0$. To find critical points $\tilde{\Delta}$ we use the necessary

conditions $J' = 0$. We use for shorter notation $\mathcal{I}$ and $\hat{S}$ instead of $\mathcal{I}\left(\Delta, \hat{S}, \hat{h}, i_{\mathrm{no}}, I, C_{\mathrm{R}}\right)$ and $\hat{S}\left(\Delta\right)$ and we start with

$$\begin{aligned}
J' =& \frac{\left(1 - \mathrm{e}^{-i_{\mathrm{no}}(\Delta+W)}\,\hat{S}\right)\left(\mathcal{I}' + C_{\mathrm{M}} i_{\mathrm{no}}\,\mathrm{e}^{-i_{\mathrm{no}}\Delta}\,\hat{S} - C_{\mathrm{M}}\,\mathrm{e}^{-i_{\mathrm{no}}\Delta}\,\hat{S}'\right)}{\left(1 - \mathrm{e}^{-i_{\mathrm{no}}(\Delta+W)}\,\hat{S}\right)^2} \\
& - \frac{\left(\mathcal{I} - C_{\mathrm{M}}\,\mathrm{e}^{-i_{\mathrm{no}}\Delta}\,\hat{S}\right)\left(i_{\mathrm{no}}\,\mathrm{e}^{-i_{\mathrm{no}}\Delta}\,\hat{S} - \mathrm{e}^{-i_{\mathrm{no}}\Delta}\,\hat{S}'\right)}{\left(1 - \mathrm{e}^{-i_{\mathrm{no}}(\Delta+W)}\,\hat{S}\right)^2} \\
=& \frac{\left(1 - \mathrm{e}^{-i_{\mathrm{no}}(\Delta+W)}\,\hat{S}\right)\left(\mathcal{I}' + C_{\mathrm{M}}\,\mathrm{e}^{-i_{\mathrm{no}}\Delta}\left(i_{\mathrm{no}}\hat{S} - \hat{S}'\right)\right)}{\left(1 - \mathrm{e}^{-i_{\mathrm{no}}(\Delta+W)}\,\hat{S}\right)^2} \\
& - \frac{\mathrm{e}^{-i_{\mathrm{no}}\Delta}\left(\mathcal{I} - C_{\mathrm{M}}\,\mathrm{e}^{-i_{\mathrm{no}}\Delta}\,\hat{S}\right)\left(i_{\mathrm{no}}\hat{S} - \hat{S}'\right)}{\left(1 - \mathrm{e}^{-i_{\mathrm{no}}(\Delta+W)}\,\hat{S}\right)^2}.
\end{aligned} \tag{3.4}$$

Since

$$\mathcal{I}' = \mathrm{e}^{-i_{\mathrm{no}}\Delta}\,\hat{S}\left(I - C_{\mathrm{R}}\hat{h}\right),$$

we obtain

$$\begin{aligned}
(3.4) =& \frac{\left(1 - \mathrm{e}^{-i_{\mathrm{no}}(\Delta+W)}\,\hat{S}\right)\left(\mathrm{e}^{-i_{\mathrm{no}}\Delta}\,\hat{S}\left(I - C_{\mathrm{R}}\hat{h}\right) + C_{\mathrm{M}}\,\mathrm{e}^{-i_{\mathrm{no}}\Delta}\left(i_{\mathrm{no}}\hat{S} - \hat{S}'\right)\right)}{\left(1 - \mathrm{e}^{-i_{\mathrm{no}}(\Delta+W)}\,\hat{S}\right)^2} \\
& - \frac{\mathrm{e}^{-i_{\mathrm{no}}\Delta}\left(\mathcal{I} - C_{\mathrm{M}}\,\mathrm{e}^{-i_{\mathrm{no}}\Delta}\,\hat{S}\right)\left(i_{\mathrm{no}}\hat{S} - \hat{S}'\right)}{\left(1 - \mathrm{e}^{-i_{\mathrm{no}}(\Delta+W)}\,\hat{S}\right)^2} \\
=& \mathrm{e}^{-i_{\mathrm{no}}\Delta}\,\frac{\left(1 - \mathrm{e}^{-i_{\mathrm{no}}(\Delta+W)}\,\hat{S}\right)\left(\hat{S}\left(I - C_{\mathrm{R}}\hat{h} + C_{\mathrm{M}} i_{\mathrm{no}}\right) - C_{\mathrm{M}}\hat{S}'\right)}{\left(1 - \mathrm{e}^{-i_{\mathrm{no}}(\Delta+W)}\,\hat{S}\right)^2} \\
& - \mathrm{e}^{-i_{\mathrm{no}}\Delta}\,\frac{\left(\mathcal{I} - C_{\mathrm{M}}\,\mathrm{e}^{-i_{\mathrm{no}}\Delta}\,\hat{S}\right)\left(i_{\mathrm{no}}\hat{S} - \hat{S}'\right)}{\left(1 - \mathrm{e}^{-i_{\mathrm{no}}(\Delta+W)}\,\hat{S}\right)^2}.
\end{aligned}$$

We use the intermediate value theorem, confer [40], to show the existence of a solution in the interval $I_S := \left[0, \tilde{\Delta}\right]$ with $\tilde{\Delta} \leq \hat{h}^{-1}\left(\frac{I}{C_{\mathrm{R}}}\right)$. The restriction to the interval I_S is not very tight, because

$$I - C_{\mathrm{R}}\hat{h}\left(\Delta\right) \geq 0 \quad \text{for all } \Delta \in I_S$$

and the gas turbine operator has got only positive expected revenue. We observe that

$$\mathcal{I}\left(\Delta\right) \geq 0 \quad \text{for all } \Delta \in I_S.$$

Further, we remark, that J is continuous in Δ which is necessary for the intermediate value theorem. Thus, we obtain

$$\begin{aligned} J'(0) &= \frac{\left(1-\mathrm{e}^{-i_{\mathrm{no}}W}\right)\left(\left(I-C_{\mathrm{R}}\hat{h}+C_{\mathrm{M}}i_{\mathrm{no}}\right)-C_{\mathrm{M}}\hat{S}'\right)+C_{\mathrm{M}}\left(i_{\mathrm{no}}-\hat{S}'\right)}{\left(1-\mathrm{e}^{-i_{\mathrm{no}}W}\right)^2} \\ &= \frac{\overbrace{\left(I-C_{\mathrm{R}}\hat{h}(0)+C_{\mathrm{M}}i_{\mathrm{no}}\right)}^{(\mathrm{c})}\overbrace{-C_{\mathrm{M}}\hat{S}'}^{>0}}{\underbrace{1-\mathrm{e}^{-i_{\mathrm{no}}W}}_{>0}}+\frac{C_{\mathrm{M}}\left(i_{\mathrm{no}}-\hat{S}'\right)}{\underbrace{\left(1-\mathrm{e}^{-i_{\mathrm{no}}W}\right)^2}_{>0}} \end{aligned} \tag{3.5}$$

If equation (3.5)(c) > 0 holds, we have $J'(0) > 0$. It is true, if $\frac{I+C_{\mathrm{M}}i_{\mathrm{no}}}{C_{\mathrm{R}}} > \hat{h}(0)$. As next step, we estimate an upper bound for $J'\left(\tilde{\Delta}\right)$. We obtain

$$\begin{aligned} J'\left(\tilde{\Delta}\right) =& \mathrm{e}^{-i_{\mathrm{no}}\tilde{\Delta}}\frac{\left(1-\mathrm{e}^{-i_{\mathrm{no}}\left(\tilde{\Delta}+W\right)}\hat{S}\right)\left(\hat{S}\left(I-C_{\mathrm{R}}\hat{h}+C_{\mathrm{M}}i_{\mathrm{no}}\right)-C_{\mathrm{M}}\hat{S}'\right)}{\left(1-\mathrm{e}^{-i_{\mathrm{no}}\left(\tilde{\Delta}+W\right)}\hat{S}\right)^2} \\ &- \mathrm{e}^{-i_{\mathrm{no}}\tilde{\Delta}}\frac{\left(\mathcal{I}-C_{\mathrm{M}}\mathrm{e}^{-i_{\mathrm{no}}\tilde{\Delta}}\hat{S}\right)\left(i_{\mathrm{no}}\hat{S}-\hat{S}'\right)}{\left(1-\mathrm{e}^{-i_{\mathrm{no}}\left(\tilde{\Delta}+W\right)}\hat{S}\right)^2} \\ =& \mathrm{e}^{-i_{\mathrm{no}}\tilde{\Delta}}\frac{\left(1-\mathrm{e}^{-i_{\mathrm{no}}\left(\tilde{\Delta}+W\right)}\hat{S}\right)\left(\hat{S}C_{\mathrm{M}}i_{\mathrm{no}}-C_{\mathrm{M}}\hat{S}'\right)}{\left(1-\mathrm{e}^{-i_{\mathrm{no}}\left(\tilde{\Delta}+W\right)}\hat{S}\right)^2} \\ &- \mathrm{e}^{-i_{\mathrm{no}}\tilde{\Delta}}\frac{\left(\mathcal{I}-C_{\mathrm{M}}\mathrm{e}^{-i_{\mathrm{no}}\tilde{\Delta}}\hat{S}\right)\left(i_{\mathrm{no}}\hat{S}-\hat{S}'\right)}{\left(1-\mathrm{e}^{-i_{\mathrm{no}}\left(\tilde{\Delta}+W\right)}\hat{S}\right)^2} \\ =& \mathrm{e}^{-i_{\mathrm{no}}\tilde{\Delta}}\left(i_{\mathrm{no}}\hat{S}-\hat{S}'\right)\frac{\left(1-\mathrm{e}^{-i_{\mathrm{no}}\left(\tilde{\Delta}+W\right)}\hat{S}\right)C_{\mathrm{M}}-\mathcal{I}+C_{\mathrm{M}}\mathrm{e}^{-i_{\mathrm{no}}\tilde{\Delta}}\hat{S}}{\left(1-\mathrm{e}^{-i_{\mathrm{no}}\left(\tilde{\Delta}+W\right)}\hat{S}\right)^2} \\ =& \underbrace{\mathrm{e}^{-i_{\mathrm{no}}\tilde{\Delta}}\left(i_{\mathrm{no}}\hat{S}\left(\tilde{\Delta}\right)-\hat{S}'\left(\tilde{\Delta}\right)\right)}_{>0}\frac{\overbrace{\left(1+\hat{S}\left(\tilde{\Delta}\right)\mathrm{e}^{-i_{\mathrm{no}}\tilde{\Delta}}\left(1-\mathrm{e}^{-i_{\mathrm{no}}W}\right)\right)}^{<2}C_{\mathrm{M}}-\overbrace{\mathcal{I}\left(\tilde{\Delta}\right)}^{>0}}{\left(1-\mathrm{e}^{-i_{\mathrm{no}}\left(\tilde{\Delta}+W\right)}\hat{S}\left(\tilde{\Delta}\right)\right)^2} \end{aligned}$$

If

$$\mathcal{I}\left(\tilde{\Delta}\right) > 2C_{\mathrm{M}} \tag{3.6}$$

holds, then $J'\left(\tilde{\Delta}\right) < 0$ follows and we have $J'(0) > 0 > J'\left(\tilde{\Delta}\right)$. The intermediate value theorem delivers the existence of at least one

$$\zeta \in I_s \quad \text{with } J'(\zeta) = 0. \tag{3.7}$$

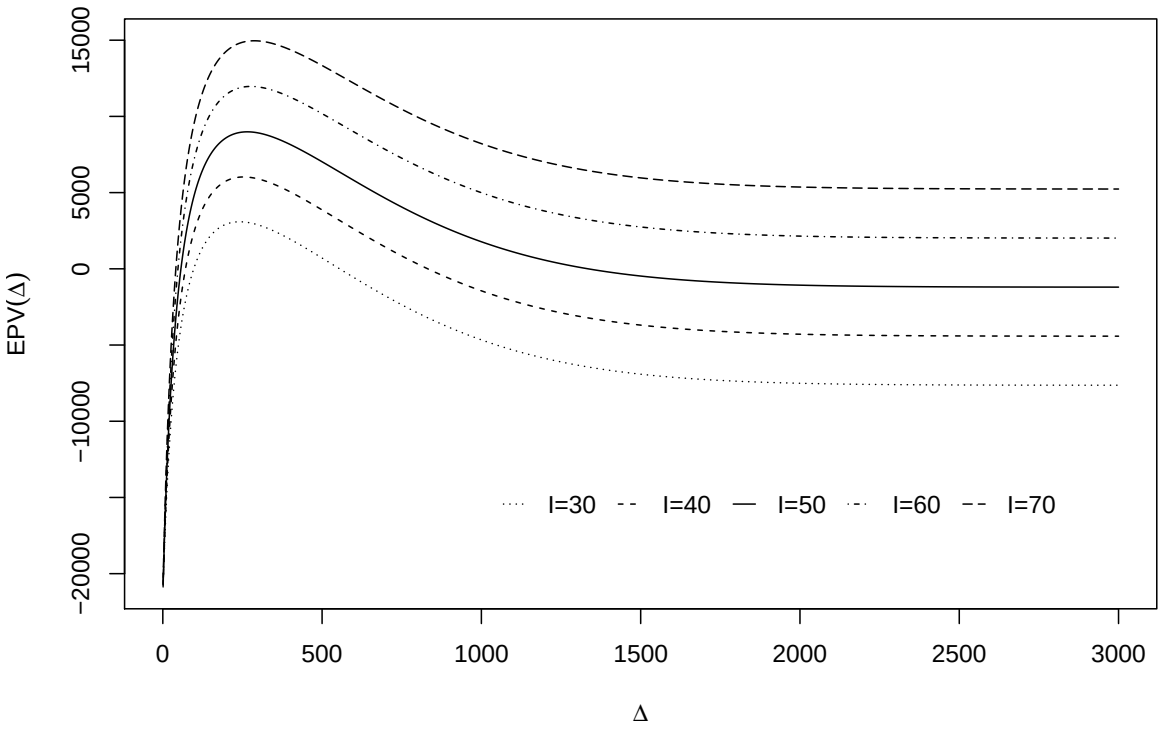


Figure 3.2.: $J(\Delta)$: Expected present value for a gas turbine with maintenance for different cash flows I.

We need the assumption in equation (3.6) to avoid the case that $\Delta = \tilde{\Delta} = \infty$ becomes the optimal solution. If $\zeta \neq 0$ and $\zeta \neq \tilde{\Delta}$, then there exists $\epsilon > 0$ that holds

$$J'(\Delta) > 0 \text{ for all } \Delta \in (\zeta - \epsilon, \zeta) \quad \text{and} \quad J'(\Delta) < 0 \text{ for all } \Delta \in (\zeta, \zeta + \epsilon). \tag{3.8}$$

Therefore, equations (3.7) and (3.8) deliver us the fulfillment of the necessary and the sufficient conditions for a maximum, confer Theorem 4.13 and Theorem 4.15 from section 4.1.3.

3.3. Example for one Component

In the following, we present a numerical example for our toy model. We consider a Weibull distribution for the hazard rate $\hat{h}(t)$ with a shape parameter m and scale parameter η for the failure probability. According to section 2.1.3, we get for the hazard rate, cumulative hazard rate and the survival function without maintenance the following representations

$$\hat{h}(t) = \frac{m}{\eta}\left(\frac{t}{\eta}\right)^{m-1}, \ \hat{H}(t) = -\left(\frac{t}{\eta}\right)^{m} \text{ and } \hat{S}(t) = \mathrm{e}^{-\left(\frac{t}{\eta}\right)^{m}}.$$

I	30	40	50	60	70
$\Delta^\star$	241	254	265	276	287
$J(\Delta^\star)$	3085.2	6025.9	8987.1	11966.1	14960.9

Table 3.1.: Results for the toy model with one component. We present the optimal interval length $\Delta^\star$ and the associated value of $J(\Delta^\star)$ for the different I.

For the associated functions with maintenance we obtain

$$h(t) = \frac{m}{\eta}\left(\frac{t \mod \Delta + W}{\eta}\right)^{m-1},$$
$$H(t) = -\left\lfloor \frac{t}{\Delta+W} \right\rfloor \left(\frac{\Delta}{\eta}\right)^m - \left(\frac{t \mod \Delta + W}{\eta}\right)^m$$
$$\text{and } S(t) = \exp\left(-\left\lfloor \frac{t}{\Delta+W} \right\rfloor \left(\frac{\Delta}{\eta}\right)^m - \left(\frac{t \mod \Delta + W}{\eta}\right)^m\right).$$

Finally, our cost functional has the following representation

$$\begin{aligned} J(\Delta) &= \frac{\mathcal{I}\left(\Delta, \hat{S}, \hat{h}, i_{\text{no}}, I, C_{\text{R}}\right) - C_{\text{M}}\, \mathrm{e}^{-i_{\text{no}}\Delta}\, \hat{S}(\Delta)}{1 - \mathrm{e}^{-i_{\text{no}}(\Delta+W)}\, \hat{S}(\Delta)} \\ &= \frac{\int_0^\Delta \mathrm{e}^{-i_{\text{no}}\tau}\, \hat{S}(\tau)\left(I - C_{\text{R}}\hat{h}(\tau)\right) d\tau - C_{\text{M}}\, \mathrm{e}^{-i_{\text{no}}\Delta}\, \hat{S}(\Delta)}{1 - \mathrm{e}^{-i_{\text{no}}(\Delta+W)}\, \hat{S}(\Delta)} \\ &= \frac{\int_0^\Delta \mathrm{e}^{-i_{\text{no}}\tau}\, \mathrm{e}^{-\left(\frac{\tau}{\eta}\right)^m}\left(I - C_{\text{R}}\frac{m}{\eta}\left(\frac{\tau}{\eta}\right)^{m-1}\right) d\tau - C_{\text{M}}\, \mathrm{e}^{-i_{\text{no}}\Delta}\, \mathrm{e}^{-\left(\frac{\Delta}{\eta}\right)^m}}{1 - \mathrm{e}^{-i_{\text{no}}(\Delta+W)}\, \mathrm{e}^{-\left(\frac{\Delta}{\eta}\right)^m}}. \end{aligned}$$

In our example we consider the following parameters: The shape and scale parameter of the Weibull distribution are

$$\eta = 2000.0 \quad \text{and} \quad m = 2.4.$$

The parameters are chosen according the data from [13] and [70]. In particular we orientated us at estimate data, [70], and actual maintenance concepts, [13]. The financial parameter are

$$I = 30, 40, \ldots, 70,\ C_{\text{M}} = 300,\ C_{\text{R}} = 500000,\ i_{\text{no}} = 0.003 \text{ and } W = 30.$$

We considered data from European Energy Exchange to determine the parameter I, confer [36].

In Section 3.3, we see $J(\Delta)$ plotted against Δ for different values $I = 30, \ldots, 70$ of the operators revenue. In Table 3.1, we see the optimal interval length $\Delta^\star$ for the different I and the associated values of the objective function J. If the revenue I doubles from 30 to 60 the overall revenue J becomes four times bigger, but the interval length growth only by 1.19. In general, the increasing revenue I has bigger effect on the optimal value of $J(\Delta^\star)$, than on the optimal interval length $\Delta^\star$.

3.4. The Extension to a two Component Model

We extend our maintenance model to a gas turbine which consists of two component groups and has got two life counters. This takes us a step closer to our problem formulation from Section 2.3.1 where we consider multiple components. For the two component case in our toy model we allow following two service actions:

1. Replace component group one or
2. Replace component group one and two.

The operating time between two service actions of type 1 is given by $\Delta_1 > 0$ and $\Delta_2 > 0$ is the operation time between two maintenance actions of type 2. The service actions are done periodically and we assume

$$\Delta_2 = \beta\Delta_1 \quad \text{for a fixed } \beta \in \mathbb{N}.$$

$W_1 > 0$ is the duration for the service action of type 1 and $W_2 > 0$ is a the additional time to do the service action of type 2. Therefore, $W_1 + W_2$ is the complete service duration of service action of type 2. C_{M_1} denotes the maintenance costs for the service action 1 and C_{M_2} denotes the additional costs to do service action 2. As before the complete maintenance cost for service action 2 are given by $C_{\mathrm{M}_1} + C_{\mathrm{M}_2}$. According to Definition 3.1 we obtain:

Definition 3.2
For $i = 1, 2$ we define:

- *Hazard rate of the i-th component without maintenance:* $\hat{h}_i(t)$
- *Cumulative hazard rate of the i-th component without maintenance:*

$$\hat{H}_i(t) := \int_0^t \hat{h}_i(\tau)\, d\tau.$$

- *Survival function of the the i-th component without maintenance:*

$$\hat{S}_i(t) := \mathrm{e}^{-\hat{H}(t)} = \mathrm{e}^{-\int_0^t \hat{h}_i(\tau) d\tau}.$$

- *Survival function of the gas turbine without maintenance (serial connection):*

$$\hat{S}(t) := \hat{S}_1(t) \cdot \hat{S}_2(t) = \mathrm{e}^{-\int_0^t \hat{h}_1(\tau) + \hat{h}_2(\tau) d\tau}.$$

- *I the operators revenue, C_R the financial risk of a gas turbine failure, C_{M_i} the maintenance cost for i-th component, i_{no} discount factor.*

- *The ki-th operating interval:*

$$\mathbb{I}_{ki} := [i(\Delta_1 + W_1) + kW_2, (i+1)\Delta_1 + iW_1 + kW_2]$$

for $k = 0, 1, \ldots,\ i = k\beta, \ldots, k\beta + \beta - 1$.

- *Overall operating time:*

$$\mathbb{I} = \bigcup_{k=0}^{\infty} \bigcup_{i=k\beta}^{k\beta+\beta-1} \mathbb{I}_{ki}.$$

As in the one component case, a service action resets the hazard rate to zero after maintenance for the respective component group. For notational simplicity, we define the following quantities: The number of outages of component 2 until t:

$$n_2 := \left\lfloor \frac{t}{\Delta_2 + \beta W_1 + W_2} \right\rfloor \tag{3.9}$$

and the number of outages of component 1 after the last outage of component 2 until t:

$$n_1 := \left\lfloor \frac{t - \left\lfloor \frac{t}{\Delta_2+\beta W_1+W_2} \right\rfloor (\Delta_2 + \beta W_1 + W_2)}{\Delta_1 + W_1} \right\rfloor = \left\lfloor \frac{t - n_2(\Delta_2 + \beta W_1 + W_2)}{\Delta_1 + W_1} \right\rfloor. \tag{3.10}$$

According to Definition 3.2 and equations (3.9) and (3.10), we get the hazard rate functions

$$h_1(t) := \hat{h}_1(t - n_2(\Delta_2 + \beta W_1 + W_2) - n_1(\Delta_1 + W_1)), \tag{3.11}$$

$$h_2(t) := \hat{h}_2(t - n_2(\Delta_2 + \beta W_1 + W_2) - n_1 W_1), \tag{3.12}$$

the cumulative hazard rate functions for the two components

$$\begin{aligned} H_1(t) &:= (\beta n_2 + n_1) \int_0^{\Delta_1} \hat{h}_1(\tau)\, d\tau + \int_0^{t-n_2(\Delta_2+\beta W_1+W_2)-n_1(\Delta_1+W_1)} \hat{h}_1(\tau)\, d\tau \\ &= (\beta n_2 + n_1) \hat{H}_1(\Delta_1) + \int_0^{t-n_2(\Delta_2+\beta W_1+W_2)-n_1(\Delta_1+W_1)} \hat{h}_1(\tau)\, d\tau, \end{aligned} \tag{3.13}$$

$$\begin{aligned} H_2(t) &:= n_2 \int_0^{\Delta_2} \hat{h}_2(\tau)\, d\tau + \int_0^{t-n_2(\Delta_2+\beta W_1+W_2)-n_1 W_1} \hat{h}_2(\tau)\, d\tau \\ &= n_2 \hat{H}_2(\Delta_2) + \int_0^{t-n_2(\Delta_2+\beta W_1+W_2)-n_1 W_1} \hat{h}_2(\tau)\, d\tau \end{aligned} \tag{3.14}$$

and the survival functions for the two components

$$S_1(t) := \exp(-H_1(t)) = \hat{S}_1(\Delta_1)^{\beta n_2 + n_1} \hat{S}_1(t - n_2(\Delta_2 + \beta W_1 + W_2) - n_1(\Delta_1 + W_1)), \tag{3.15}$$

$$S_2(t) := \exp(-H_2(t)) = \hat{S}_2(\Delta_2)^{n_2} \hat{S}_2(t - n_2(\Delta_2 + \beta W_1 + W_2) - n_1 W_1). \tag{3.16}$$

Now, we can use the expected cash flow derivation from the Section 3.1 and we obtain

$$\begin{aligned}\mathrm{EPV} = \int_0^\infty \mathrm{e}^{-i_{\mathrm{no}}\tau}\, S_1(\tau)\, S_2(\tau) \big(I\chi_{\{\tau\in\mathbb{I}\}} &- C_{\mathrm{R}} \cdot (h_1(\tau) + h_2(\tau)) \\ &- C_{\mathrm{M}_1}\delta_{\{\tau \mod (\Delta_1+W_1+\beta W_2)\}} \\ &- C_{\mathrm{M}_2}\delta_{\{\tau \mod (\Delta_1+W_1+W_2)\}} \big)\, d\tau. \end{aligned} \tag{3.17}$$

for expected discountend cashflow. Further, we can split equation (3.17) with the help of equations (3.11) to (3.16) and the definition of the operating time $\mathbb{I}$ into three parts: The revenue and risk part

$$\sum_{k=0}^{\infty} \sum_{i=k\beta}^{k\beta+\beta-1} \int_{i(\Delta_1+W_1)+kW_2}^{(i+1)\Delta_1+iW_1+kW_2} \mathrm{e}^{-i_{\mathrm{no}}\tau}\, S_1(\tau)\, S_2(\tau) \left(I - C_{\mathrm{R}} \cdot (h_1(\tau) + h_2(\tau))\right) d\tau, \tag{3.18}$$

the service cost for component 1

$$\begin{aligned} -\sum_{k=0}^{\infty} C_{\mathrm{M}_2}\, \mathrm{e}^{-i_{\mathrm{no}}((k+1)\Delta_2+k\beta W_1+kW_2)} &S_1\left((k+1)\Delta_2 + k\beta W_1 + kW_2\right) \\ &\cdot S_2\left((k+1)\Delta_2 + k\beta W_1 + kW_2\right) \end{aligned} \tag{3.19}$$

and the service cost for component 2

$$\begin{aligned} -\sum_{i=\beta k}^{\beta k-\beta-1} C_{\mathrm{M}_1}\, \mathrm{e}^{-i_{\mathrm{no}}((i+1)\Delta_1+iW_1+kW_2)} &S_1\left(((i+1)\Delta_1 + iW_1 + kW_2)\right) \\ &\cdot S_2\left((i+1)\Delta_1 + iW_1 + kW_2\right). \end{aligned} \tag{3.20}$$

To simplify equation (3.18) we use the same arguments for the geometric series as in Section 3.1. We obtain

$$\begin{aligned} &\sum_{k=0}^{\infty} \sum_{i=0}^{\beta-1} \int_{(i+k\beta)(\Delta_1+W_1)+kW_2}^{\Delta_1+(i+k\beta)W_1+kW_2} \mathrm{e}^{-i_{\mathrm{no}}\tau}\, S_1(\tau)\, S_2(\tau) \left(I - C_{\mathrm{R}} \cdot (h_1(\tau) + h_2(\tau))\right) d\tau \\ =&\sum_{k=0}^{\infty} \sum_{i=0}^{\beta-1} \int_0^{\Delta_1} \mathrm{e}^{-i_{\mathrm{no}}(\tau+(i+k\beta)(\Delta_1+W_1)+kW_2)}\, \hat{S}_1(\tau)^{\beta k+i}\, \hat{S}_1(\tau)\, \hat{S}_2(\tau)^k\, \hat{S}_2(\tau + i\Delta_1) \\ &\cdot \left(I - C_{\mathrm{R}} \cdot \left(\hat{h}_1(\tau) + \hat{h}_2(\tau + i\Delta_1)\right)\right) d\tau \\ =&\sum_{k=0}^{\infty} \mathrm{e}^{-i_{\mathrm{no}}(k\beta(\Delta_1+W_1)+kW_2)}\, \hat{S}_2(\tau)^k\, \hat{S}_1(\tau)^{\beta k} \sum_{i=0}^{\beta-1} \mathrm{e}^{-i_{\mathrm{no}}i(\Delta_1+W_1)}\, \hat{S}_1(\tau)^i \\ &\cdot \int_0^{\Delta_1} \mathrm{e}^{-i_{\mathrm{no}}\tau}\, \hat{S}_1(\tau)\, \hat{S}_2(\tau + i\Delta_1) \left(I - C_{\mathrm{R}} \left(\hat{h}_1(\tau) + \hat{h}_2(\tau + i\Delta_1)\right)\right) d\tau \\ =&\frac{\sum_{i=0}^{\beta-1} \mathrm{e}^{-i_{\mathrm{no}}i(\Delta_1+W_1)}\, \hat{S}_1(\tau)^i \int_0^{\Delta_1} \mathrm{e}^{-i_{\mathrm{no}}\tau}\, \hat{S}_1(\tau)\, \hat{S}_2(\tau + i\Delta_1)\, d\tau}{1 - \mathrm{e}^{-i_{\mathrm{no}}(\beta(\Delta_1+W_1)+kW_2)}\, \hat{S}_2(\tau)\, \hat{S}_1(\tau)^\beta} \end{aligned}$$

$$\cdot \left(I -_{\mathrm{R}} \left(\hat{h}_1 (\tau) + \hat{h}_2 (\tau + i\Delta_1) \right) \right)$$
$$=: \frac{\mathcal{I}\left(\Delta_1, \beta, \hat{S}_1, \hat{S}_2, \hat{h}_1, \hat{h}_2, i_{\mathrm{no}}, I, C_{\mathrm{R}}, W_1, W_2\right)}{1 - \mathrm{e}^{-i_{\mathrm{no}}(\beta(\Delta_1+W_1)+kW_2)} \hat{S}_2 (\tau) \hat{S}_1 (\tau)^{\beta}}.$$

As next step we simplify equation (3.20) minus equation (3.19)

$$(3.20) - (3.19) =$$
$$= \sum_{k=0}^{\infty} C_{\mathrm{M}_2} \mathrm{e}^{-i_{\mathrm{no}}((k+1)\Delta_2 + k\beta W_1 + kW_2)}$$
$$\cdot S_1 ((k+1)\Delta_2 + k\beta W_1 + kW_2) S_2 ((k+1)\Delta_2 + k\beta W_1 + kW_2)$$
$$- \sum_{i=0}^{\beta-1} C_{\mathrm{M}_1} \mathrm{e}^{-i_{\mathrm{no}}((i+\beta k+1)\Delta_1 + (i+\beta k)W_1 + kW_2)} S_1 ((i+1+\beta k)\Delta_1 + (i+\beta k) W_1 + kW_2)$$
$$\cdot S_2 ((i+1+\beta k)\Delta_1 + (i+\beta k) W_1 + kW_2)$$
$$= \sum_{k=0}^{\infty} C_{\mathrm{M}_2} \mathrm{e}^{-i_{\mathrm{no}}((k+1)\Delta_2 + k\beta W_1 + kW_2)} \hat{S}_1^{\beta k+\beta} (\Delta_1) \hat{S}_2^{k+1} (\Delta_2)$$
$$- \sum_{i=0}^{\beta-1} C_{\mathrm{M}_1} \mathrm{e}^{-i_{\mathrm{no}}((i+\beta k+1)\Delta_1 + (i+\beta k)W_1 + kW_2)} \hat{S}_1^{i+1+\beta k} (\Delta_1) \hat{S}_2^{k} (\Delta_2) \hat{S}_2 ((i+1)\Delta_1)$$
$$= \sum_{k=0}^{\infty} \mathrm{e}^{-i_{\mathrm{no}}(k\Delta_2 + k\beta W_1 + kW_2)} \hat{S}_1^{\beta k} (\Delta_1) \hat{S}_2^{k} (\Delta_2)$$
$$\cdot \left(C_{\mathrm{M}_2} \mathrm{e}^{-i_{\mathrm{no}}\Delta_2} \hat{S}_1^{\beta} (\Delta_1) \hat{S}_2 (\Delta_2) - \sum_{i=0}^{\beta-1} C_{\mathrm{M}_1} \mathrm{e}^{-i_{\mathrm{no}}((i+1)\Delta_1 + iW_1)} \hat{S}_1^{i+1} (\Delta_1) \hat{S}_2 ((i+1)\Delta_1) \right)$$
$$= \frac{C_{\mathrm{M}_2} \mathrm{e}^{-i_{\mathrm{no}}\Delta_2} \hat{S}_1^{\beta} (\Delta_1) \hat{S}_2 (\Delta_2) - \sum_{i=0}^{\beta-1} C_{\mathrm{M}_1} \mathrm{e}^{-i_{\mathrm{no}}((i+1)\Delta_1 + iW_1)} \hat{S}_1^{i+1} (\Delta_1) \hat{S}_2 ((i+1)\Delta_1)}{1 - \mathrm{e}^{-i_{\mathrm{no}}(\beta(\Delta_1+W_1)+W_2)} \hat{S}_1^{\beta} (\Delta_1) \hat{S}_2 (\Delta_2)}$$
$$=: \frac{\mathcal{C}_{\mathcal{M}}\left(\Delta_1, \beta, \hat{S}_1, \hat{S}_2, \hat{h}_1, \hat{h}_2, i_{\mathrm{no}}, I, C_{\mathrm{R}}, W_1, W_2\right)}{1 - \mathrm{e}^{-i_{\mathrm{no}}(\beta(\Delta_1+W_1)+W_2)} \hat{S}_1^{\beta} (\Delta_1) \hat{S}_2 (\Delta_2)}.$$

Summarizing, we have the following representation for the expected present value

$$J(\Delta_1, \beta) = \frac{\mathcal{I}\left(\Delta_1, \beta, \hat{S}_1, \hat{S}_2, \hat{h}_1, \hat{h}_2, i_{\mathrm{no}}, I, C_{\mathrm{R}}, W_1, W_2\right)}{1 - \mathrm{e}^{-i_{\mathrm{no}}(\beta(\Delta_1+W_1)+W_2)} \hat{S}_1^{\beta} (\Delta_1) \hat{S}_2 (\Delta_2)}$$
$$- \frac{\mathcal{C}_{\mathcal{M}}\left(\Delta_1, \beta, \hat{S}_1, \hat{S}_2, \hat{h}_1, \hat{h}_2, i_{\mathrm{no}}, I, C, W_1, W_2\right)}{1 - \mathrm{e}^{-i_{\mathrm{no}}(\beta(\Delta_1+W_1)+W_2)} \hat{S}_1^{\beta} (\Delta_1) \hat{S}_2 (\Delta_2)}$$

and our optimization problem for the two component case is given by

$$\max_{\Delta_1, \beta} J(\Delta_1, \beta)$$
$$\text{subject to } \beta \in \mathbb{N},$$
$$\Delta_1 \geq 0.$$

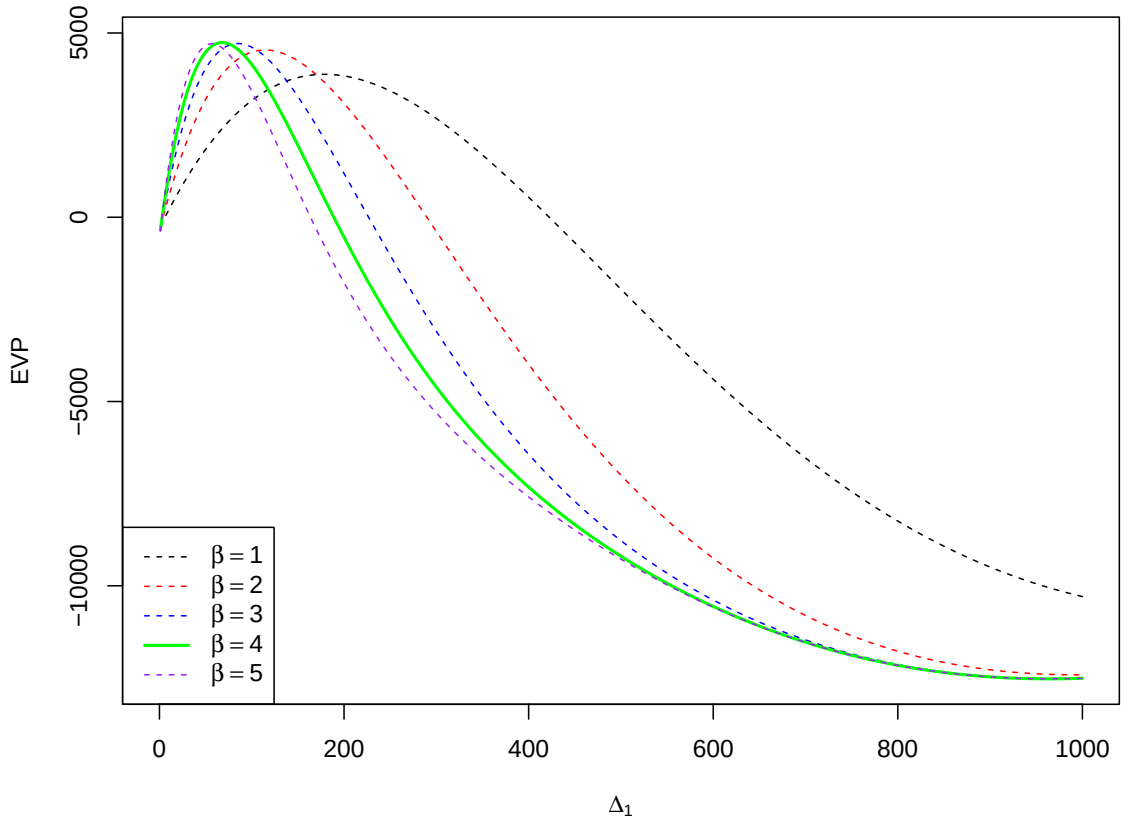


Figure 3.3.: $J(\Delta_1, \beta)$: Expected present value for a gas turbine with predictive maintenance which consists of two components. The bold green line refers to the optimal choice of $\beta = 4$.

3.5. Example for two Components

In the following, we present an example for the two component toy model. We consider a Weibull distribution for the hazard rate $\hat{h}_i(t)$, $i = 1, 2$, with shape parameters $m_1 = 1.4$ and $m_2 = 1.1$ and scale parameters $\eta_1 = 2000$ and $\eta_2 = 2200$. In particular we obtain for the representations of the hazard rates and the survival functions without maintenance the following form:

$$\hat{h}_1(t) = \frac{2.4}{2000}\left(\frac{t}{2000}\right)^{1.4}, \quad \hat{h}_2(t) = \frac{2.1}{2200}\left(\frac{t}{2200}\right)^{1.1},$$
$$\hat{S}_1(t) = \mathrm{e}^{-\left(\frac{t}{2000}\right)^{2.4}} \quad \text{and} \quad \hat{S}_2(t) = \mathrm{e}^{-\left(\frac{t}{2200}\right)^{2.1}}.$$

The financial model parameters are summarized in table 3.2 and they are based on the same sources as the parameters in Section 3.3.

i_{no}	I	C_{R}	W_1	W_2	C_{M_1}	C_{M_2}
0.003	50	500000	20	10	100	200

Table 3.2.: Financial model parameters for two component toy model.

Then we obtain for the solution:

$$\max_{\Delta} J(\Delta_1, \beta) \approx 4740.588 = J(68.4, 4)$$

and in Figure 3.3 we see $J(\Delta_1, \beta)$ plotted against Δ_1 for different values of $\beta = 1, \ldots, 5$. Despite the small difference in the Weibull parameters, $\eta_2/\eta_1 = 1.1$ and $m_2/m_1 = 0.875$, there is a big difference $\beta = 4$ in the interval length for the two components. It should be a result of the higher maintenance cost C_{M_2} of component two.

4. Replacement Model

The task of the toy model in Chapter 3 was to present in a simple way the principle in our modeling approach to calculate and optimize the expected value of a discounted cash flow. To get a more realistic model, we extend the toy model to a replacement model.

The new model has a lot of advantages when compared to the toy model. We can include an arbitrary number of parts in the model and we do not need the restriction that the outage times are integer multiplies of each other as requested in the toy model in Section 3.4. Further, the time horizon is finite, in contrast to the toy model. As a further improvement, all model parameters like the financial data can be time dependent. There are no restrictions to the outage pattern and outage combination. We include the dismantle dependencies of the gas turbine parts to get a more realistic model and model the advantage of combining service actions. In addition the operating regime of the gas turbine can be flexible and need not be constant. But we keep the restriction that a failed component won't be replaced and that every service action sets the corresponding components to an "as new" condition.

This chapter is organized as follows: First, we give a short introduction to optimal control and we extend it to impulse control. Impulse control theory is the basis of our solution approach. Further, we present necessary and sufficient optimality conditions for this type of optimal control problems. Then we transform the service outage problem into a solvable nonlinear optimization problem. We introduce definitions and ideas to model our replacement problem in terms of impulse control. In the last section, we present numerical results for a model problem.

4.1. Optimal Control Theory

Optimal control theory is an extension or a generalization of calculus of variations theory. Since 1696 the attention in calculus of variations has grown. Johann Bernoulli postulated his famous Brachistochrone-Problem to various mathematician like Newton and Leibniz. For a more detailed description confer [71]. Optimal control belongs to a class of optimization problems to get control polices and it is infinite dimensional. The development was motivated by military applications like ballistic trajectory optimization since 1950. Further important applications can be found in test drive simu-

lation, robotic control or business management. The important results were delivered by Pontryagin and his students in [72]. They postulated and proved necessary optimal conditions for optimal control problems. Hestenes proved independently the same results, confer [49]. In 1977 Blaquiere proved necessary optimal conditions for control problems with impulse dynamics, confer [12]. Chahim used this approach to optimize the dike heights, confer [21]. In [23] the authors presented a maintenance model for machine replacement which is less complex than our model. It does not support multiple components, disassembly constraints or failure probabilities.

In the next part we present the basic results of optimal control problems. We are interested in the dynamical development of a state x. The state x can belong to a mechanic or economic system or problem which can be described with a (partial) differential equation. In general we can influence such a system by a control u. It is possible that there are boundary conditions for the state x or control u, because there are, for example, physical restrictions present. We are not interested in all feasible states and controls, we are searching the optimal pair of control and state which minimize a given objective function $J(\cdot,\cdot)$. In summary we have the following elements of a control problem, confer [43]:

- The state variables $x(t) \in \mathbb{R}^{n_x}$ at time t.
- The control variables $u(t) \in \mathbb{R}^{n_u}$ at time t.
- The differential equation $\dot{x}(t) = g(x(t), u(t), t) \in \mathbb{R}^{n_x}$ for the system dynamics.
- The objective function

$$J(u(t), x(t)) = \mathrm{e}^{-i_{\mathrm{no}}T} G_0(x(0), x(T)) + \int_0^T \mathrm{e}^{-i_{\mathrm{no}}t} G(x(t), u(t), t)\, dt$$

  which is dived into the terminal cost G_0 and the continuous reward G.
- The mixed restrictions $\Phi(x(t), u(t), t) \in \mathbb{R}^{n_\Phi}$.
- The state restrictions $x(t) \in \Omega_x(t) \subseteq \mathbb{R}^{n_x}$.
- The control restrictions $u(t) \in \Omega_u(t) \subseteq \mathbb{R}^{n_u}$.

In general the optimal control problem has got the following form and it is illustrated in Figure 4.1:

Problem 4.1 (General Optimal Control)
Find a control $u(t)$ and a state $x(t)$ which solve

$$\max_{u,x} \mathrm{e}^{-i_{no}T} G_0(x(T)) + \int_0^T \mathrm{e}^{-i_{no}t} G(x(t), u(t), t)\, dt$$

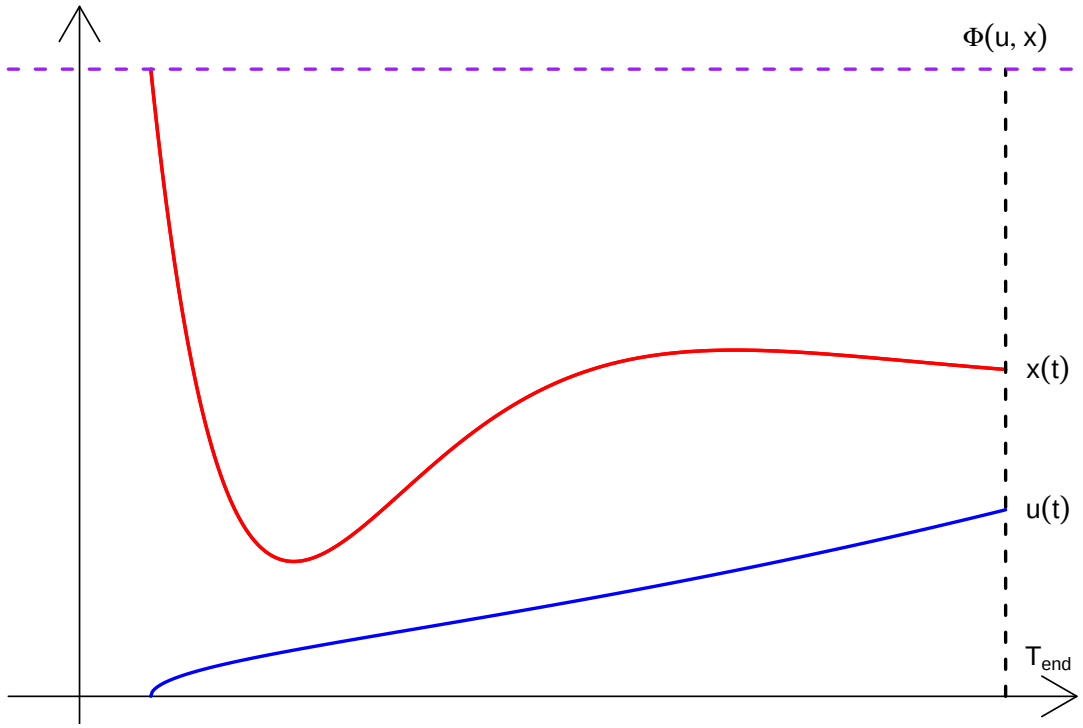


Figure 4.1.: Schematic illustration of the general control Problem 4.1. The red line represents the state x and the blue line the control u.

subject to

$$\begin{aligned}
\dot{x}(t) &= g(x(t), u(t), t), \quad 0 \leq t \leq T && \textit{(dynamics)} \\
x(0) &= x_0, && \textit{(boundary restriction)} \\
\Phi(x(t), u(t), t) &\leq 0, \quad 0 \leq t \leq T, && \textit{(mixed restriction)} \\
x(t) &\in \Omega_x, \quad 0 \leq t \leq T, && \textit{(state restriction)} \\
u(t) &\in \Omega_u, \quad 0 \leq t \leq T. && \textit{(control restriction)}
\end{aligned}$$

Next, we give an academical example of a rocket car to illustrate the formulation of Problem 4.1 and to introduce a special kind of solution which will be important later in this thesis. Our problem is to reach a target destination as fast as possible with a rocket car. We can control the thrust $u \in [u_{\min}, u_{\max}]$ of the rocket car. Further, the initial position x_0 is given and newtons second law of motion gives us the state dynamics by

$$\begin{aligned}
u(t) &= m\ddot{x}(t) \\
x(0) &= x_0
\end{aligned}$$

with the car mass m. In summary the control problem is given by

$$\min_u \int_0^T 1dt = \min_u T \quad \text{(minimize the time)}$$

subject to

$$\begin{aligned} u(t) &= m\ddot{x}(t) \\ x(0) &= x_0,\ x(T) = 0 \\ \dot{x}(T) &= 0 \\ -1 = u_{\min} \leq & u(t) \leq u_{\max} = 1. \end{aligned}$$

Here, the optimal control $u^\star$ is

$$u^\star(t) = \begin{cases} -\operatorname{sign}(x_0), & t \in [0, T^\star/2] \\ \operatorname{sign}(x_0), & t \in [T^\star/2, T^\star] \end{cases}$$

with the minimal and optimal time $T^\star$. Such kind of a solution is of type "bang-bang", because the control takes only values on the boundaries of Ω_u. It will become important in our maintenance problem to reduce the complexity of the solution. We discuss it in Section 4.1.1 and Section 4.4 again. The following section delivers us basic theoretical results about optimal control problems and especially about impulse control problems.

4.1.1. Theoretical Results

We deliver for simplicity only results for optimal control problem with mixed state and control restriction and without explicit state and control constraints. This distinction is only important from a theoretical point of view. Results for the general case are presented in [79] and [91].

The Maximum Principle

We start with the definition of the Hamiltonian function of our control problem by

$$\mathcal{H}\text{am}(t, x(t), u(t), \lambda(t)) = \lambda(t)^{\mathrm{T}} g(x(t), u(t), t) + G(x(t), u(t), t)$$

where $\lambda(t) \in \mathbb{R}^{n_x}$ is a vector function of costate variables. Further, we define the Lagrangian $\mathcal{L}$ by

$$\mathcal{L}(t, x(t), u(t), \lambda(t), \mu(t)) = \mathcal{H}\text{am}(t, x(t), u(t), \lambda(t)) + \mu(t)^{\mathrm{T}} \Phi(x(t), u(t), t)$$

The Hamiltonian gives us important information for the necessary optimal condition and Pontryagin used it to deliver his minimal principle which is also known as the maximum principle, confer [15], [41], [72] and [91]:

Theorem 4.2 (Maximum Principle)
Let the following condition be true for the optimal control problem from Problem 4.1:

1. *The functions G_0, G, g, Φ are continuous regarding all arguments and continuously differentiable in x and u.*

2. *Let $(x^\star, u^\star)$ be a maximum of the optimal control problem with mixed constraints.*

3. *The optimal solution $(x^\star, u^\star)$ satisfies the constraint qualifications, namely that the $n_\Phi \times (n_\Phi + 1)$-matrix*

$$\begin{pmatrix} \frac{\partial}{\partial u}\Phi_1\left(x\left(t\right), u\left(t\right), t\right) & \Phi_1\left(x\left(t\right), u\left(t\right), t\right) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial}{\partial u}\Phi_{n_\Phi}\left(x\left(t\right), u\left(t\right), t\right) & 0 & \dots & \Phi_{n_\Phi}\left(x\left(t\right), u\left(t\right), t\right) \end{pmatrix}$$

   has, along the optimal solution $(x^\star, u^\star)$, full rank for every $t \in [0, T]$.

4. *There are no state or control restrictions.*

Then there exist a continuous and piecewise continuously differentiable function $\lambda\left(\cdot\right) \in \mathbb{R}^{n_g}$ and a piecewise continuous multiplier function $\mu\left(\cdot\right) \in \mathbb{R}^{n_\Phi}$ satisfying the following conditions at every time t where $u^\star$ is continuous:

1. ***Adjoint differential equation:***

$$\dot{\lambda}\left(t\right) = i_{no}\lambda\left(t\right) - \frac{\partial}{\partial x}\mathcal{L}\left(t, x^\star\left(t\right), u^\star\left(t\right), \lambda\left(t\right), \mu\left(t\right)\right), \tag{4.1}$$

2. ***Transversality condition:***

$$\lambda\left(T_{end}\right) = \frac{\partial}{\partial x}G_0\left(x^\star\left(T\right)\right), \tag{4.2}$$

3. ***Complementary slackness condition:***

$$\sigma\left(t\right) \geq 0, \quad \sigma\left(t\right)^T \Phi\left(x^\star\left(t\right), u^\star\left(t\right), t\right) = 0 \tag{4.3}$$

4. ***Optimality condition:***

$$\mathcal{H}am\left(t, x^\star\left(t\right), u^\star\left(t\right), \lambda\left(t\right)\right) = \max_{u(\cdot)} \mathcal{H}am\left(t, x^\star\left(t\right), u\left(t\right), \lambda\left(t\right)\right), \tag{4.4}$$

$$\frac{\partial}{\partial u}\mathcal{L}\left(t, x^\star\left(t\right), u^\star\left(t\right), \lambda\left(t\right), \sigma\left(t\right)\right) = 0. \tag{4.5}$$

The optimal condition in equation (4.4) can be changed to:

$$\mathcal{H}\text{am}\left(t, x^\star\left(t\right), u^\star\left(t\right), \lambda\left(t\right)\right) \geq \mathcal{H}\text{am}\left(t, x^\star\left(t\right), u\left(t\right), \lambda\left(t\right)\right) \quad \text{for all } u \in \Omega_u. \tag{4.6}$$

In this case the optimal control $u^\star$, as mentioned before, maximizes the Hamilton function $\mathcal{H}\text{am}$. The optimal control function $u^\star$ is characterized as an implicit function of the state $x^\star$ and the adjoint variables $\lambda^\star$. For a fixed time point t equations (4.4) and (4.6) represent an finite dimensional optimization problem. We can use finite dimensional optimization results for necessary and sufficient conditions and associated algorithms to solve the problem.

Bang-Bang Control

In this subsection we consider a special class of optimal control problems where the problem depends linearly of the control $u(t)$. The optimal control $\hat{u}$ of this class of problems has got a special structure which is very important for our modeling approach. But first, we define the new problem class and for simplicity we assume $n_u = 1$:

Problem 4.3 (Optimal control problem with linear control)
Find a continuous control $u(t)$ and a state $x(t)$ which solves

$$\max_{x,u} \mathrm{e}^{-i_{no}T} G_0\left(x\left(T_{end}\right)\right) + \int_{T_0}^{T_{end}} \mathrm{e}^{-i_{no}t} \left(G_1\left(t, x\left(t\right)\right) + G_2\left(t, x\left(t\right)\right) u\left(t\right)\right) dt$$

subject to

$$\begin{aligned} &\dot{x}\left(t\right) = g_1\left(t, x\left(t\right)\right) + g_2\left(t, x\left(t\right)\right) u\left(t\right) \quad \textit{for all } t \in \left(T_0, T_{end}\right), \\ &x\left(0\right) = x_0, \\ &u\left(t\right) \in \mathcal{U} = \left[u_{min}, u_{max}\right] \quad \textit{for all } t \in \left[T_0, T_{end}\right]. \end{aligned}$$

We divide the cost function G from our general Problem 4.1 into two parts G_1 and G_2, because G is linear in u and we can write it as $G = G_1 + G_2 u$. With the same argumentation we did the breakdown of g. For further notation we define the function $\Gamma\colon \left[T_0, T_{\text{end}}\right] \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \mapsto \mathbb{R}$ by

$$\Gamma\left(t, x, \lambda\right) := G_2\left(t, x\right) + \lambda^{\mathrm{T}} g_2\left(t, x\right) \tag{4.7}$$

and name it "shifting function" of Problem 4.3. We use the maximum principle from Theorem 4.2 and we obtain

$$\mathcal{H}\text{am}\left(t, x, u, \lambda\right) = G_1\left(t, x\right) + \lambda^{\mathrm{T}} g_1\left(t, x\right) + \Gamma\left(t, x, \lambda, \right) u$$

for the Hamilton function $\mathcal{H}\text{am}$, if we use the shifting function. From the maximum principle we obtain

$$u^\star\left(t\right) = \arg\max_{u \in \mathcal{U}} \mathcal{H}\text{am}\left(t, x, u, \lambda\right) = \arg \max_{u_{\min} \le u \le u_{\max}} \Gamma\left(t, x, \lambda\right) u.$$

Since $\mathcal{H}$am is linear in u, confer [5], we obtain the result for the optimal control $u^\star$ directly, if $\Gamma \neq 0$:

$$u^\star(t) = \begin{cases} u_{\min}, & \text{if } \Gamma(t,x,\lambda) < 0 \\ u_{\max}, & \text{if } \Gamma(t,x,\lambda) > 0 \\ \text{unknown}, & \text{if } \Gamma(t,x,\lambda) = 0 \end{cases}. \tag{4.8}$$

Equation (4.8) gives us the information that the minimum principle cannot determine the optimal control $u^\star$, if the shifting function vanishes in a time interval $[t_1, t_2]$. We use this observation for an accurate definition:

Definition 4.4 (Bang-Bang-Control and Singular Control)
Let u be a linear control and $[t_1, t_2] \subseteq [T_0, T_{end}]$.

- *The control u is named Bang-Bang-control in $[t_1, t_2]$, if Γ has got only isolated roots. The isolated roots are called shifting points.*
- *The control u is named singular control in $[t_1, t_2]$, if $\Gamma \equiv 0$ in $[t_1, t_2]$.*

In Figure 4.2 we show exemplary a bang-bang and a singular control with the corresponding shifting function. If all roots of Γ are isolated, then we can use equation (4.8) to determine the control directly. The shifting points t_i are implicit given through the condition

$$\Gamma(t,x,\lambda) = 0.$$

4.1.2. Impulse Control

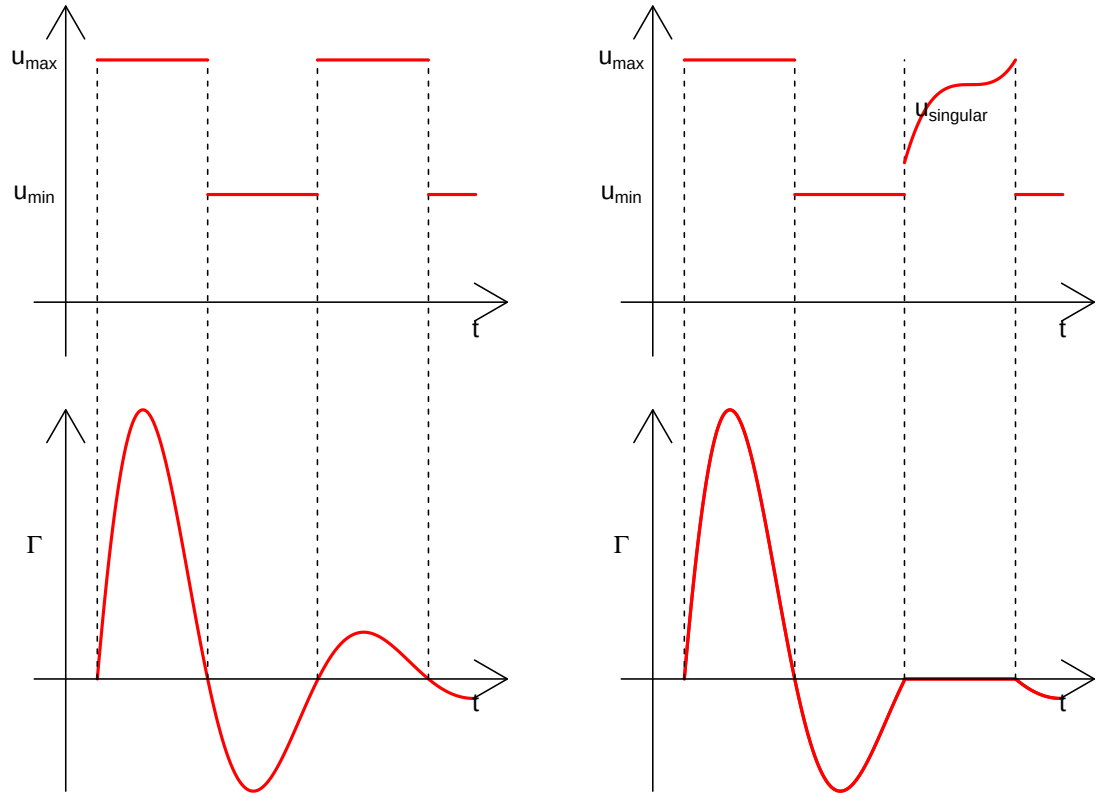
Here we present an extension of the optimal control approach which allows and controls jumps in our our state variable $x(t)$. We follow closely the ideas from [12]. The general formulation of the impulse control Problem is:

Problem 4.5 (General Formulation of Impulse Control)
Find a continuous control $u(t)$, a number N of jump points $\tau \in \mathbb{R}^N$, $\tau_i < \tau_{i+1}$, and jump control $v \in \mathbb{R}^N$ which solves

$$\max_{N,u,v,\tau} \int_0^T \mathrm{e}^{-i_{no}t} G(x(t), u(t), t)\, dt + \sum_{i=1}^{N} \mathrm{e}^{-i_{no}\tau_i} G_I\left(x\left(\tau_i^-\right), v^i, \tau_i^-\right) + \mathrm{e}^{-i_{no}T} G_0(x(T))$$

subject to

$$\begin{aligned} x(0) &= x_0 \\ \dot{x}(t) &= g(x(t), u(t), t), \ t \notin \{\tau_1, \ldots, \tau_N\} \\ x\left(\tau_i^+\right) - x\left(\tau_i^-\right) &= g_I\left(x\left(\tau_i^-\right), v^i, \tau_i^-\right), \ i = 1, \ldots, N \\ x \in \Omega_x, u \in \Omega_u, v \in \Omega_v, \tau_i &\in [0, T). \end{aligned}$$



(a) Bang bang control without singular sections and the corresponding shifting function.

(b) Bang bang control with singular sections and the corresponding shifting function.

Figure 4.2.: Bang Bang and singular controls with the corresponding shifting functions. The upper part shows the control u and the lower part the corresponding shifting function Γ.

In this case x is the state variable, u is a control variable and v^i is the impulse control variable. The functions $x(t)$ and $u(t)$ are piecewise continuous. The future rewards are discounted by the discount factor $e^{-i_{no}t}$ with the discount rate i_{no}. The number of jumps is denoted by N and τ_i is the time moment of the i-th jump or impulse of the system. τ_j^+ and τ_j^- are the time points just after and before τ_j. The terminal time of the system is $T > 0$. The income or revenue of the system is given by $G(x(t), u(t), t)$ and the cost of the i-th jump is represented by $G_I(x(t), v^i, t)$. $G_0(x(T^+))$ is the salvage value. Figure 4.3 illustrates an impulse control example schematically. The continuous change of the state variable $x(t)$ is described by $g(x(t), u(t), t)$ and $g_I(x(t), v^i, t)$ denotes the instantaneous change of the state variable at an impulse jump. Further, we assume that the domains $\Omega_u \subset \mathbb{R}^{n_u}$ and $\Omega_v \subset \mathbb{R}^{n_v}$ are bounded convex sets.

Furthermore, we extend our definition of the Hamiltonian function by the present value

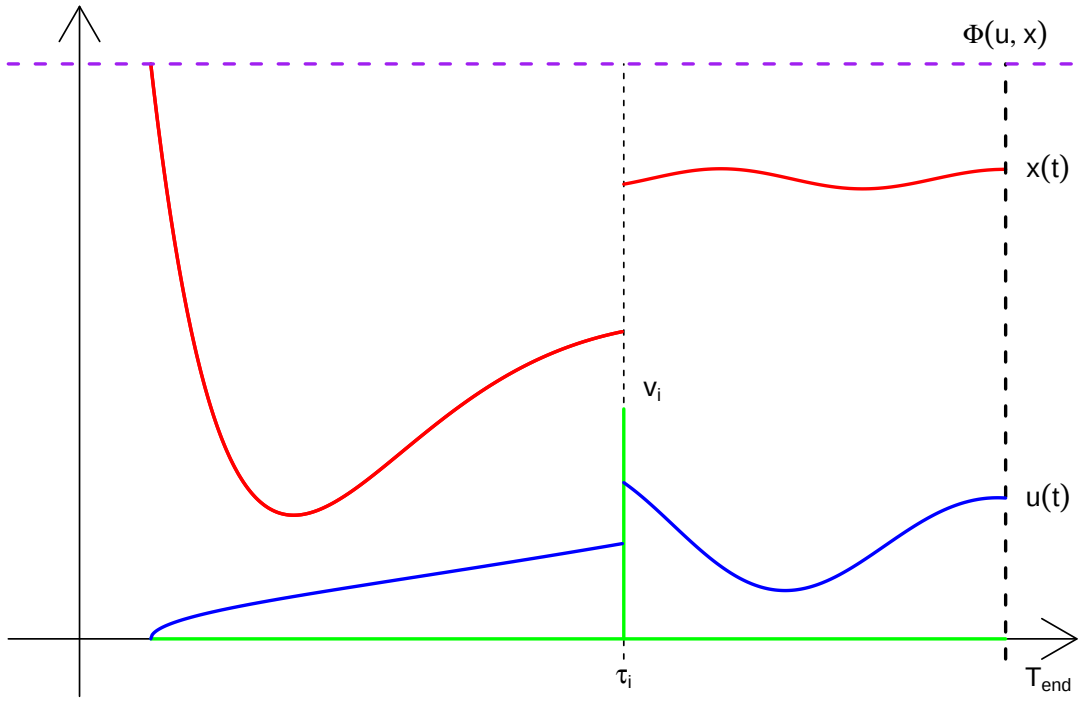


Figure 4.3.: Schematic illustration of impulse control Problem 4.5 with a jump at τ_i. The red curve represents the state x, the blue curve the continuous control u and the green one the impulse control v_i.

Hamiltonian function formulation given by

$$\mathcal{H}\mathrm{am}\left(t, x\left(t\right), u\left(t\right), \lambda\left(t\right)\right) := \mathrm{e}^{-i_{\mathrm{no}}t}\, G\left(x\left(t\right), u\left(t\right), t\right) + \lambda\left(t\right) g\left(x\left(t\right), u\left(t\right)\right)$$

and we establish the present value impulse Hamiltonian by

$$\mathcal{IH}\mathrm{am}\left(t, x\left(t\right), v^i, \lambda\left(t\right)\right) := \mathrm{e}^{-i_{\mathrm{no}}t}\, G_{\mathrm{I}}\left(x\left(t\right), v^i, t\right) + \lambda\left(t\right) g_{\mathrm{I}}\left(x\left(t\right), v^i, t\right).$$

Necessary and Sufficiency Conditions

Now, we deliver necessary conditions for the general impulse control Problem 4.5 which was delivered by Blaquiere in [12]. But we take the formulation from [24] to be consistent in notation.

Theorem 4.6 (Impulse Control Necessary Conditions)
Let $\left(x^\star\left(t\right), u^\star\left(t\right), N^\star, \tau_1^\star, \ldots, \tau_N^\star, v^{1\star}, \ldots, v^{N\star}\right)$ *be an optimal solution for the impulse control problem defined in Problem 4.5. Then there exists an adjoint variable* λ *such*

that the following conditions hold:

$$u^{\star}(t) = \arg\max_{u \in \Omega_u} \mathcal{H}am\left(t, x^{\star}(t), u(t), \lambda(t)\right) \tag{4.9}$$

$$\dot{\mu}(t) = -\frac{\partial}{\partial x}\mathcal{H}am\left(t, x^{\star}(t), u(t), \lambda(t)\right) \tag{4.10}$$

At the jump points, it is true that

$$\frac{\partial}{\partial v^i}\mathcal{IH}am\left(\tau_i^{\star}, x^{\star}\left(\tau_i^{+\star}\right), v^i, \lambda\left(\tau_i^{+\star}\right)\right)\left(v^i - v^{i\star}\right) \leq 0 \quad \textit{for all } v^i \in \Omega_v \tag{4.11}$$

$$\lambda\left(\tau_i^{+\star}\right) - \lambda\left(\tau_i^{-\star}\right) = -\frac{\partial}{\partial x}\mathcal{IH}am\left(\tau_i^{-\star}, x^{\star}\left(\tau_i^{-\star}\right), v^{i\star}, \lambda\left(\tau_i^{-\star}\right)\right) \tag{4.12}$$

$$\begin{aligned}&\mathcal{H}am\left(\tau_i^{\star}, x^{\star}\left(\tau_i^{+\star}\right), u^{\star}\left(\tau_i^{+\star}\right), \lambda\left(\tau_i^{+\star}\right)\right) - \mathcal{H}am\left(\tau_i^{\star}, x^{\star}\left(\tau_i^{-\star}\right), u^{\star}\left(\tau_i^{-\star}\right), \lambda\left(\tau_i^{-\star}\right)\right)\\ &- \frac{\partial}{\partial \tau}\mathcal{IH}am\left(\tau_i^{\star}, x^{\star}\left(\tau_i^{\star -}\right), v^{i\star}, \lambda\left(\tau_i^{-\star}\right)\right)\begin{cases} > 0 & \textit{for } \tau_i^{\star} = 0\\ = 0 & \textit{for } \tau_i^{\star} \in (0,T)\\ < 0 & \textit{for } \tau_i^{\star} = T.\end{cases}\end{aligned} \tag{4.13}$$

For all points in time at which there is no jump, i.e. $t \neq \tau_i$, $i = 1, \ldots, k$,

$$\frac{\partial}{\partial v^i}\mathcal{IH}am\left(x^{\star}(t), 0, \lambda(t)\right) v^i \leq 0. \tag{4.14}$$

At the horizon date T^+ the transversality condition

$$\lambda\left(T^+\right) = \mathrm{e}^{i_{no}T^+}\frac{\partial}{\partial x}G_0\left(x^{\star}\left(T^+\right)\right) \tag{4.15}$$

holds.

We present a proof for Theorem 4.6 only for the case that $G\left(x(t), u(t), t\right) = 0$ and g does not depends on $u(t)$. Further, the state dynamic functions g and g_{I} do not depend on t. We follow the ideas from [77] strictly. The assumption of the proof give us the opportunity to use the concept of "needle variation" from [39] to prove the theorem. The "needle variation" concept is an often used method to prove necessary conditions in optimal control, confer [51] and therefore more general. It uses the idea of a variation of a curve, we will produce variations of trajectories and controls with this method. A needle variation changes the value of the control instantaneously to a constant value over a closed interval of specified length. This will let us study how changing the control affects the system.

To follow the proof from [77], we further need the following two assumptions:

A1: The functions $g(\cdot)$, G_0 and $g_{\mathrm{I}}(\cdot, v^j,)$ are continuously differentiable on $\mathbb{R}^n$ for every $v^j \in \Omega_v$,

A2: For every $x \in \Omega_x$ the set $\{g_{\mathrm{I}}(x\colon\ v^j),\ v^j \in \Omega_v\}$ is convex.

Now, we are able to present the proof:

Proof. The proof starts with the definition of the needle variations of our control, state and terminal reward. Then, we show that partial derivatives of needle variation of the terminal reward exists. As last step, we present that this partial derivatives of the needle variations imply the optimal conditions of our theorem.

Let $v \in \Omega_v$, $j \leq N$ and $\epsilon > 0$ be such that

$$\tau_j + \epsilon < \tau_{j+1} \quad \text{and} \quad \tau_j - \epsilon > \tau_{j-1}.$$

From assumption **A2**, the convexity of g_{I} in v^j, it follows that for arbitrary $\beta \in [0,1]$ there exists a $v(\beta) \in \Omega_v$ such that

$$g_{\mathrm{I}}\left(x(\tau_j), v^j\right) + \beta\left(g_{\mathrm{I}}(x(\tau_k), v) - g_{\mathrm{I}}\left(x(\tau_j), v^j\right)\right) = g_{\mathrm{I}}(x(\tau_k), v(\beta)).$$

Without loss of generality we may assume

$$v(0) = v^j \quad \text{and} \quad v(1) = v.$$

Let $\alpha \in (\tau_j - \epsilon, \tau_j + \epsilon)$. Then, we define:

$$\begin{aligned}
\pi^{j-1}(\alpha,\beta) &:= \left((\tau_1,\ldots,\tau_{j-1}), \left(v^1,\ldots,v^{j-1}\right)\right), \\
\pi^{j}(\alpha,\beta) &:= \left((\tau_1,\ldots,\tau_{j-1},\alpha), \left(v^1,\ldots,v^{j-1},v(\beta)\right)\right), \\
\pi^{k}(\alpha,\beta) &:= \left((\tau_1,\ldots,\tau_{j-1},\alpha,\tau_{j+1},\ldots,\tau_k), \left(v^1,\ldots,v^{j-1},v(\beta),v^{j+1},\ldots,v^k\right)\right), \\
&\quad k = j+1,\ldots,N.
\end{aligned}$$

Here, π defines a possible solution / policy of our impulse control problem. We vary the optimal control v^j in a small area. The parameter α gives us the variation around τ_j and β presents the variation in the control v^j. These variations in the control lead to the name "needle variation". Let

$$x^k(t;\alpha,\beta) = x^{\pi^k(\alpha,\beta)}(t) \quad \text{for } t \in [0,T],\ k = j-1,\ldots,N,$$

be the resulting state trajectory to policy π. In particular it is the solution of the boundary value problem in the impulse control problem where β is the boundary value at time α. We remark that

$$x^N(t;\alpha,\beta) = x(t),\ t \in [0,T].$$

We obtain for arbitrary $\alpha \in (\tau_j - \epsilon, \tau_j + \epsilon)$ that

$$\sigma(\alpha,\beta) := G_0\left(x^N(T;\alpha,\beta)\right) \leq G_0\left(x^N(T;\tau_j,0)\right) = G_0(x(T)).$$

Therefore, the function σ has got first order partial derivatives at $(\tau_j, 0)$ and we obtain

$$\frac{\partial}{\partial\alpha}\sigma(\tau_j,0) = 0, \quad \text{if } \tau_j > 0, \tag{4.16}$$

$$\frac{\partial}{\partial\alpha}\sigma(\tau_j,0) \leq 0, \quad \text{if } \tau_j = 0, \tag{4.17}$$

$$\frac{\partial}{\partial\beta}\sigma(\tau_j,0) \leq 0, \quad \text{for all } \tau_j. \tag{4.18}$$

The main idea of the proof is to show that equations (4.16) to (4.18) imply the condition in equations (4.11) and (4.13) of Theorem 4.6. The conditions in equations (4.16) to (4.18) transform under the proof's assumptions to

$$\begin{aligned} \lambda\left(\tau_i^+\right)\left(g_{\mathrm{I}}\left(x\left(\tau^i\right), v^i\right) - g_{\mathrm{I}}\left(x\left(\tau^i\right), v\right)\right) &\geq 0, \quad \text{for } v \in \Omega_v, && (4.19)\\ \lambda\left(\tau_i^+\right) g\left(x\left(\tau_i^+\right)\right) - \lambda\left(\tau_i\right) g\left(x\left(\tau_i\right)\right) &= 0, \quad \text{if } \tau_j > 0, && (4.20)\\ &\geq 0, \quad \text{if } \tau_j = 0. \end{aligned}$$

We start to prove that equation (4.19) holds. We define

$$\xi(\beta) := \sigma(\tau_j, \beta) = G_0\left(x^N\left(T; \tau_j, \beta\right)\right), \quad \beta \in [0,1].$$

From the chain rule we obtain that

$$\begin{aligned} \frac{\partial}{\partial\beta}\xi(\beta) &= \frac{\partial}{\partial x} G_0\left(x^N\left(T; \tau_j, \beta\right)\right) \frac{\partial}{\partial\beta} x^N\left(T; \tau_j, \beta\right) \\ &= \frac{\partial}{\partial x} G_0\left(x^N\left(T; \tau_j, \beta\right)\right) \frac{\partial}{\partial\beta} x\left(T; \tau_j, \delta_N(\beta)\right) \end{aligned} \tag{4.21}$$

with $\delta_N(\beta) = x^{N-1}\left(\tau_N; \tau_j, \beta\right) + g_{\mathrm{I}}\left(x^{N-1}\left(\tau_N; \tau_j, \beta\right), v^1\right)$. Further, we remark that the function

$$\lambda(s) = \lambda\left(\tau_j\right) Q\left(\tau_j, s\right), \quad s \in [\tau_j, \tau_{j+1}]$$

fulfills

$$\dot{\lambda} = -\lambda \frac{\partial}{\partial x} g(x) \text{ on } [\tau_j, \tau_{j+1}] \quad \text{and} \quad p\left(\tau_j^+\right) = \lambda\left(\tau_{j+1}^+\right) Q\left(\tau_{j+1}, \tau_j^+\right) \tag{4.22}$$

for arbitrary $j = 1, \ldots, T$ and with the definition

$$Q(t,s) := \frac{\partial}{\partial x} g\left(t, s; x(s)\right), \quad t, s \in [0,T].$$

The remarks follows from classical results of ordinary differential equation theory. We conclude with the help of the remark for the function ξ:

$$\begin{aligned} \frac{\partial}{\partial\beta}\xi(0) = \lambda(T) Q\left(T, \tau_N^+\right) \Bigg(& \frac{\partial}{\partial\beta} x^{N-1}\left(\tau_N; \tau_j, \beta\right)|_{\beta=0} \\ & + \frac{\partial}{\partial x} g_{\mathrm{I}}\left(x^{N-1}\left(\tau_N; \tau_j, \beta\right), v^N\right) \frac{\partial}{\partial\beta} x^{N-1}\left(\tau_N; \tau_j, \beta\right)|_{\beta=0} \Bigg). \end{aligned}$$

Further, we obtain with the help of the remark in equation (4.22) and equation (4.12)

$$\begin{aligned} \lambda(T) Q\left(T, \tau_N^+\right) &= \lambda\left(\tau_N^+\right) \\ \lambda\left(\tau_N^+\right) g_{\mathrm{I}}\left(x^{N-1}\left(\tau_N; \tau_j, 0\right), v^N\right) &= \lambda\left(\tau_N\right) - \lambda\left(\tau_N^+\right). \end{aligned}$$

Hence, the partial derivative of $\xi(0)$ fulfills

$$\begin{aligned} \frac{\partial}{\partial\beta}\xi(0) &= \lambda\left(\tau_N^+\right) \frac{\partial}{\partial\beta} x^{N-1}\left(\tau_N; \tau_j, \beta\right)|_{\beta=0} + \left(\lambda\left(\tau_N\right) - \lambda\left(\tau_N^+\right)\right) \frac{\partial}{\partial\beta} x^{N-1}\left(\tau_N; \tau_j, \beta\right)|_{\beta=0} \\ &= \lambda\left(\tau_N\right) \frac{\partial}{\partial\beta} x^{N-1}\left(\tau_N; \tau_j, \beta\right)|_{\beta=0} \end{aligned}$$

With a simple induction argument we get

$$\frac{\partial}{\partial\beta}\xi\left(0\right)=\lambda\left(\tau_{j+1}\right)\frac{\partial}{\partial\beta}x^{j}\left(\tau_{j+1};\tau_{j},\beta\right)|_{\beta=0}.$$

Similar to equation (4.21) and the help of assumption **A1** we obtain

$$x^{j}\left(\tau_{j+1};\tau_{j},\beta\right)=x\left(\tau_{j+1},\tau_{j},\delta_{j}\left(\beta\right)\right)$$

with

$$\begin{aligned}\delta_{j}\left(\beta\right)&=x^{j-1}\left(\tau_{j};\tau_{j},\beta\right)+g_{\mathrm{I}}\left(x^{j-1}\left(\tau_{j};\tau_{j},\beta\right),v\left(\beta\right)\right)\\&=x\left(\tau_{j}\right)+g_{\mathrm{I}}\left(x\left(\tau_{j}\right),v^{j}\right)+\beta\left(g_{\mathrm{I}}\left(x\left(\tau_{j}\right),v\right)-g_{\mathrm{I}}\left(x\left(\tau_{j}\right),v^{j}\right)\right).\end{aligned}$$

Therefore, we obtain for the partial derivative

$$\begin{aligned}\frac{\partial}{\partial\beta}\xi\left(0\right)&=\lambda\left(\tau_{j+1}\right)Q\left(\tau_{j+1},\tau_{j}^{+}\right)\left(g_{\mathrm{I}}\left(x\left(\tau_{j}\right),v\right)-g_{\mathrm{I}}\left(x\left(\tau_{j}\right),v^{j}\right)\right)\\&=\lambda\left(\tau_{j}^{+}\right)\left(g_{\mathrm{I}}\left(x\left(\tau_{j}\right),v\right)-g_{\mathrm{I}}\left(x\left(\tau_{j}\right),v^{j}\right)\right).\end{aligned}$$

From this point equation (4.19) follows, because we have

$$\frac{\partial}{\partial\beta}\sigma\left(\tau_{j},0\right)=\frac{\partial}{\partial\beta}\xi\left(0\right)\leq 0.$$

As next step we prove equation (4.20). We define for an arbitrary $\alpha\in\left(\tau_{j}-\epsilon,\tau_{j}+\epsilon\right)$

$$\Psi\left(\alpha\right)=\ G_{0}\left(x^{N}\left(T;\alpha,0\right)\right).$$

With the help of the chain rule we obtain

$$\begin{aligned}\frac{\partial}{\partial\alpha}\Psi\left(\alpha\right)&=\frac{\partial}{\partial x}G_{0}\left(x^{N}\left(T;\alpha,0\right)\right)\frac{\partial}{\partial\alpha}x^{N}\left(T;\alpha,0\right)\\&=\frac{\partial}{\partial x}G_{0}\left(x^{N}\left(T;\alpha,0\right)\right)\frac{\partial}{\partial\alpha}x\left(T,\tau_{j};\gamma_{N}\left(\alpha\right)\right)\end{aligned}$$

with $\gamma_{N}\left(\alpha\right)=x^{N-1}\left(\tau_{N};\alpha,0\right)+g_{\mathrm{I}}\left(x^{N-1}\left(\tau_{N};\alpha,0\right)\right)$. Further, we obtain

$$\begin{aligned}\frac{\partial}{\partial\alpha}\Psi\left(\tau_{j}\right)=\lambda\left(T\right)Q\left(T,\tau_{N}^{+}\right)\Big(&x^{N-1}\left(\tau_{N};\alpha,0\right)|_{\alpha=\tau_{j}}\\&+\frac{\partial}{\partial x}g_{\mathrm{I}}\left(x^{N-1}\left(\tau_{N};\tau_{j},0\right),v^{j}\right)\frac{\partial}{\partial\alpha}x^{N-1}\left(\tau_{N};\alpha,0\right)|_{\alpha=\tau_{j}}\Big).\end{aligned}$$

We conclude with the help of the remark in equation (4.22) about the function Q and equation (4.12) that

$$\begin{aligned}\frac{\partial}{\partial\alpha}\Psi\left(\tau_{j}\right)&=\lambda\left(\tau_{N}^{+}\right)\frac{\partial}{\partial\alpha}x^{N-1}\left(\tau_{N};\alpha,0\right)|_{\alpha=\tau_{j}}+\left(\lambda\left(\tau_{N}\right)-\mu\left(\tau_{N}^{+}\right)\right)\frac{\partial}{\partial\alpha}x^{N-1}\left(\tau_{N};\alpha,0\right)|_{\alpha=\tau_{j}}\\&=\lambda\left(\tau_{N}\right)\frac{\partial}{\partial\alpha}x^{N-1}\left(\tau_{N};\alpha,0\right)|_{\alpha=\tau_{j}}.\end{aligned}$$

As before an induction argument delivers:

$$\frac{\partial}{\partial\alpha}\Psi\left(\tau_j\right)=\lambda\left(\tau_{j+1}\right)\frac{\partial}{\partial\alpha}x^j\left(\tau_{j+1};\alpha,0\right)|_{\alpha=\tau_j}.$$

By definition we get

$$x^j\left(\tau_{j+1};\alpha,0\right)=x\left(\tau_{j+1},\alpha;\gamma_j\left(\alpha\right)\right)$$

with

$$\gamma_j\left(\alpha\right)=x^{j-1}\left(\alpha;\alpha,0\right)+g_{\mathrm{I}}\left(x^{j-1}\left(\alpha;\alpha,0\right),v^j\right).$$

Therefore, we obtain

$$\begin{aligned}\frac{\partial}{\partial\alpha}\Psi\left(\tau_j\right)=&\lambda\left(\tau_{j+1}\right)\left(-g\left(x\left(\tau_{j+1},\tau_j;x\left(\tau_j\right)+g_{\mathrm{I}}\left(x\left(\tau_j\right),v^j\right)\right)\right)\right.\\&\left.+Q\left(\tau_{j+1},\tau_j^+\right)\left(g\left(x\left(\tau_j\right)\right)+\frac{\partial}{\partial x}g_{\mathrm{I}}\left(x\left(\tau_j\right),v^j\right)g\left(x\left(\tau_j\right)\right)\right)\right)\\=&-\lambda\left(\tau_{j+1}\right)g\left(x\left(\tau_{j+1}\right)\right)+\lambda\left(\tau_j^+\right)g\left(x\left(\tau_j\right)\right)\\&+\lambda\left(\tau_j^+\right)\frac{\partial}{\partial x}g_{\mathrm{I}}\left(x\left(\tau_j\right),v^j\right)g\left(x\left(\tau_j\right)\right)\end{aligned}$$

and equation (4.12) leads to

$$\begin{aligned}\frac{\partial}{\partial\alpha}\Psi\left(\tau_j\right)=&-\lambda\left(\tau_{j+1}\right)g\left(x\left(\tau_{j+1}\right)\right)+\lambda\left(\tau_j^+\right)g\left(x\left(\tau_j\right)\right)+\left(\lambda\left(\tau_j\right)-\lambda\left(\tau_j^+\right)\right)g\left(x\left(\tau_k\right)\right)\\=&-\lambda\left(\tau_{j+1}\right)g\left(y\left(\tau_{j+1}\right)\right)+\lambda\left(\tau_j\right)g\left(x\left(\tau_j\right)\right).\end{aligned}$$

We obtain

$$\lambda\left(\tau_j^+\right)g\left(x\left(\tau_j^+\right)\right)=\lambda\left(\tau_{j+1}^+\right)g\left(x\left(\tau_{j+1}\right)\right),$$

because the function $t\mapsto\lambda\left(t\right)g\left(x\left(t\right)\right)$ is constant on $\left(\tau_j,\tau_{j+1}\right]$. Finally, we get

$$\frac{\partial}{\partial\alpha}\Psi\left(\tau_j\right)=\lambda\left(\tau_j\right)g\left(x\left(\tau_j\right)\right)-\lambda\left(\tau_j^+\right)g\left(x\left(\tau_j^+\right)\right). \tag{4.23}$$

We obtain the condition in equation (4.19) by combining equation (4.23) with equations (4.16) and (4.17). □

We remark that the work in [77] presents an approach to weak the assumption about the objective function and system dynamics to prove a more general case.

Furthermore, Blaquiere presented and proved in [12] sufficient conditions for the impulse control problem. As before we take the formulation from [24]:

Theorem 4.7 (Impulse Control Sufficiency Conditions)
Let there be a feasible solution $\left(x^\star\left(t\right),u^\star\left(t\right),N^\star,\tau_1^\star,\ldots,\tau_N^\star,v^{1\star},\ldots,v^{N\star}\right)$*, for the impulse control Problem 4.5 and a piecewise continuous trajectory, so that the necessary optimality conditions of Theorem 4.6 hold. When the maximized Hamiltonian function*

$$\mathcal{H}am=\max_{u(\cdot)}\mathcal{H}am\left(t,x,u,\lambda\right)$$

is concave in x *for all* $\left(\lambda\left(t\right),t\right)$*, the* $\mathcal{IH}am$ *is concave in* (x,v) *for all* t *and* $S\left(x\right)$ *is concave in* x*, then that solution* $\left(x^\star\left(t\right),u^\star\left(t\right),N^\star,\tau_1^\star,\ldots,\tau_{N^\star}^\star,v^{1\star},\ldots,v^{N\star}\right)$*, is optimal.*

4.1.3. Nonlinear Optimization

For later use, we establish a short overview about some basic definitions and results from nonlinear programming (NLP) which we need for our solution algorithm in Section 4.2. The algorithm uses nonlinear optimization techniques. In General, a NLP is given by

$$\begin{aligned} &\max_{x \in \mathbb{R}^n} J(x) \\ &\text{subject to } \tilde{G}(x) = 0 \\ &\qquad\qquad \tilde{H}(x) \geq 0, \end{aligned} \tag{NLP}$$

where $J\colon \mathbb{R}^n \mapsto \mathbb{R}$ is the objective function, $\tilde{G}(x) = 0$ with $\tilde{G}\colon \mathbb{R}^n \mapsto \mathbb{R}^{n_{\tilde{G}}}$ are equality constraints and $\tilde{H}(x) \geq 0$ with $\tilde{H}\colon \mathbb{R}^n \mapsto \mathbb{R}^{n_{\tilde{H}}}$ are inequality constraints. For simplicity, we assume that all functions are two times continuously derivable on a sufficiently big subset $D \subset \mathbb{R}^n$. We start with elementary definitions which are taken from [1]:

Definition 4.8 (Feasible Set)
The set

$$\mathcal{F} := \left\{ x \in \mathbb{R}^n | \tilde{G}(x) = 0,\ \tilde{H}(x) \geq 0 \right\}$$

is called the feasible set of problem (NLP). A point $x \in \mathcal{F}$ is named feasible point. The problem (NLP) itself is called feasible, if $\mathcal{F} \neq \emptyset$.

Further, we need the definition of a local and global maximum:

Definition 4.9 (Local and global Maximum)
Let $\mathcal{F}$ be the feasible set of (NLP).

- *A feasible point $x^\star \in \mathcal{F}$ is a local maximum point or local maximum, if there exits a $r > 0$, such that $J(x^\star) \geq J(x)$ holds for all $x \in \mathcal{F} \cap B_r(x^\star)$, where $B_r(x^\star)$ is a Ball with radius r and midpoint $x^\star$. If we have $J(x^\star) > J(x)$ for all $x \in \mathcal{F} \cap B_r(x^\star) \setminus \{x^\star\}$, then $x^\star$ is a strict local maximum.*
- *A feasible point $x^\star \in \mathcal{F}$ is a global maximum point, if $J(x^\star) \geq J(x)$ holds for all $x \in \mathcal{F}$. If we have $J(x^\star) > J(x)$ for all $x \in \mathcal{F} \setminus \{x^\star\}$, then $x^\star$ is a strict global maximum.*

We denote the gradient of the objective function by $\nabla J(x)$. In our case it is a column vector and $\nabla \tilde{G}(x) \in \mathbb{R}^{n \times n_{\tilde{G}}}$ as well as $\nabla \tilde{H}(x) \in \mathbb{R}^{n \times n_{\tilde{H}}}$ are the transposed Jacobian matricies of $\tilde{G}$ and $\tilde{H}$. With this notation we can define the active set of inequalities in $\bar{x}$:

Definition 4.10 (Active set)
Let $\mathcal{F}$ be the feasible set of (NLP). A inequality $\tilde{H}_i(\bar{x}) \geq 0$, $1 \leq 1 \leq n_{\tilde{H}}$, is called active at point $\bar{x} \in \mathcal{F}$, if $\tilde{H}_i(\bar{x}) = 0$ holds. The set $I(\bar{x}) := \left\{ 1 \leq i \leq n_{\tilde{H}} | \tilde{H}_i(\bar{x}) = 0 \right\}$ is called the active set of point $\bar{x}$.

The next definition gives us the important notion of linear independence constraint qualification (LICQ) which we need later on.

Definition 4.11 (Regular point and LICQ)
Let $\bar{x} \in \mathcal{F}$ be a feasible point. $\bar{x}$ is called regular, if the joint Jacobian matrix of the active inequalities in $\bar{x}$ has got full row rank,

$$\operatorname{rank} \begin{pmatrix} \nabla \tilde{G}(\bar{x})^T \\ \nabla \tilde{H}_{I(\bar{x})}(\bar{x})^T \end{pmatrix} = n_{\tilde{G}} + |I(\bar{x})|$$

This condition is called linear independence constraint qualification (LICQ).

As the last definitions, we establish the Lagrange function, the complementary condition, strictly active inequalities and the tangent cone:

Definition 4.12 (Lagrangian, complementary condition, strictly active inequalities, tangent cone)

- *The function*
$$\mathcal{L}(x, \lambda, \mu) = J(x) - \lambda^T \tilde{G}(x) - \mu^T \tilde{H}(x)$$
is called Lagrangian function of problem (NLP). The vectors $\lambda \in \mathbb{R}^{N_{\tilde{G}}}$ and $\mu \in \mathbb{R}^{N_{\tilde{H}}}$ are called Lagrange multipliers associated to the equalities and inequalities restrictions.

- *The triple $(\bar{x}, \lambda, \mu)$ satisfies the complementary condition, if $\mu^T \tilde{h}(\bar{x}) \geq 0$. We obtain, then $\mu_i = 0$ or $\tilde{H}_i(\bar{x}) = 0$ or both for all $1 \leq i \leq n_{\tilde{H}}$. In the case, we either have strictly $\mu_i = 0$ or $\tilde{H}_i(\bar{x}) = 0$, the triple $(\bar{x}, \lambda, \mu)$ satisfies the strict complementary condition.*

- *An inequality $\tilde{H}_i(\bar{x}) \geq 0$ with Lagrange multiplier μ_i is called strictly active at point $\bar{x}$, if $\tilde{H}_i(\bar{x}) = 0$ and $\mu_i > 0$ hold. The associate index set of the strict active inequalities is given by*
$$I^+(\bar{x}) := \left\{ 1 \leq i \leq n_{\tilde{H}} | \tilde{H}_i(\bar{x}) \wedge \mu_i > 0 \right\}.$$

- *Let $\bar{x} \in \mathcal{F}$ be a feasible point of problem (NLP). The set*
$$\mathcal{T}(\bar{x}) := \left\{ p \in \mathbb{R}^n | \nabla \tilde{G}_i(\bar{x})^T p \geq,\ i \in I(\bar{x}) \right\},$$
is named the tangent cone.

We remark that $I^+(\bar{x}) = I(\bar{x})$, if we obtain strict complementary. Now, we describe the optimality conditions from [53] to characterize the optimal solution. We start with first order necessary conditions:

Theorem 4.13 (First order necessary conditions)
Le $x^\star \in \mathcal{F}$ be a feasible point and a local maximum of problem (NLP). Further, let $x^\star$ be a regular point. Then there are Lagrange multipliers $\lambda^\star \in \mathbb{R}^{n_{\tilde{G}}}$ and $\mu^\star \in \mathbb{R}^{n_{\tilde{H}}}$ such that

$$\begin{aligned} \nabla \mathcal{L}(x^\star, \lambda^\star, \mu^\star) &= \nabla J(x^\star) - \nabla \tilde{G}(x^\star)\lambda^\star - \nabla \tilde{H}(x^\star)\mu^\star = 0 && \text{(stationarity)} \\ {\mu^\star}^T \tilde{H}(x^\star) &= 0 && \text{(complementarity)} \\ \mu^\star &\geq 0 && \text{(dual feasibility)} \end{aligned}$$

This necessary condition is called Karush-Kuhn-Tucker-Condition or in short KKT-Condition by their discoverers. Further, every local maximum satisfies this condition, but also local minima and saddle points do. Therefore, we call the triple $(x^\star, \lambda^\star, \mu^\star)$ a stationary point. With the help of the Hessian matrix $\nabla^2 \mathcal{L}$ of the Lagrange function $\mathcal{L}$ we can deduce second order necessary conditions:

Theorem 4.14 (Second order necessary conditions)
Let the triple $(x^\star, \lambda^\star, \mu^\star)$ be a stationary point and let $x^\star$ be regular point and a local maximum. Then the following holds

$$p^T \nabla^2 \mathcal{L}(x^\star, \lambda^\star, \mu^\star)\, p \geq 0 \quad \textit{for all } p \in \mathcal{T}(x^\star).$$

From Theorem 4.14 we can conclude that the Hessian matrix of the Lagrange function $\mathcal{L}$ is positive definite. As last result we present a sufficient condition of second order:

Theorem 4.15 (Second order sufficient conditions)
Let the triple $(x^\star, \lambda^\star, \mu^\star)$ be a stationary point and $x^\star$ be regular. Further, we assume that the strict complementary condition and

$$p^T \nabla^2 \mathcal{L}(x^\star, \lambda^\star, \mu^\star)\, p > 0 \quad \textit{for all } p \in \mathcal{T}(x^\star) \setminus \{0\}$$

hold. Then $x^\star$ is a strict local maximum.

The proofs can be taken from [53, Chapter 6]. This results can be used as theoretical background for optimization algorithm like IPOPT (**I**nterior **P**oint **OPT**imizer)) which will be used in our work, confer [92].

4.2. The Direct Solution Method

To solve the optimal control problem we use "the direct method" or "gradient based method", confer [22]. The direct method uses a suitable approximation of the state variables x and control variable u and v. This discretization strategy consists of three main parts:

1. **Parametrization of the control:**

   We choose for the control u a suitable function approximation $u_{\hat{h}}$. For example we can choose a piecewise linear function on a grid. The index $\hat{h}$ refers to the mesh size of the underlying discretization / grid.

2. **Discretization of the differential equation:**

   We have to choose a suitable discretization method for the differential equation

   $$\dot{x}(t) = g(t, x(t), u(t)),$$

   e.g. the Euler-Method.

3. **Optimization:**

   After the discretization of the control and state we obtain a finite and possibly nonlinear optimization problem. We need to choose a suitable optimization algorithm, e.g. an interior point method, to solve the discretized problem.

In general, the three parts mentioned have to be chosen problem specifically, because they have to reflect the problem characteristics. To illustrate the main idea behind the method we consider Problem 4.5. We select a mesh

$$\mathbb{M}_{\hat{h}} := \{t_0 = 0 < t_1 < \ldots < t_T = T_{\text{end}}\}$$

with step sizes $\hat{h}_k = t_{k+1} - t_k$, $k = 0, \ldots, T-1$ and mesh width

$$\hat{h} := \max_{k=0,\ldots,T-1} \hat{h}_k.$$

For the discretization of the differential equation we use Euler's method, confer [99] or [28]. We start with the forward finite difference method to derive Euler's method. We get

$$\begin{aligned} \dot{x}(t) &\approx \frac{x\left(t+\hat{h}\right) - x(t)}{\hat{h}} \\ \Rightarrow x\left(t+\hat{h}\right) &= x(t) + \hat{h}\dot{x}(t) \\ \Rightarrow x\left(t+\hat{h}\right) &= x(t) + \hat{h}g(t, x(t), u(t)) \end{aligned}$$

If we use the grid points, then we obtain

$$x(t_{k+1}) = x(t_k) + \hat{h}_k g(t_k, x(t_k), u(t_k)).$$

We will use the abbreviation

$$x_k := x(t_k) \quad \text{and} \quad u_k := u(t_k).$$

to make the notation simpler. We get

$$x_{k+1} = x_k \hat{h}_k + g\left(t_k, x_k, u_k\right) \quad \text{and} \quad x_0 = \hat{x}_0. \tag{4.24}$$

Equation (4.24) is Euler's method to give us an approximation of the ordinary differential equation (ODE). For the control variable u approximation $u_{\hat{h}}$ we use a piecewise constant function $u_{\hat{h}}$ which is defined as

$$u_{\hat{h}}\left(t\right) := \begin{cases} u_0, & t = t_0 \\ u_k, & t \in \left(t_k, t_{k+1}\right],\ k = 1, \ldots, T \end{cases}.$$

As next step we define

$$\tau_i^- = t_{k-1},\ \tau_i^+ = t_k \quad \text{and} \quad \tau_i = t_k.$$

We obtain for our jump cost function

$$G_{\mathrm{I}}\left(x\left(\tau_i^-\right), v^i, \tau_i^-\right) = G_{\mathrm{I}}\left(x_{k-1}, v^k, t_k\right).$$

Further, the discrete jump dynamics are

$$x_k - x_{k-1} = g_{\mathrm{I}}\left(x_{k-1}, v^k, t_{k-1}\right).$$

We claim the fulfillment of the constraints only at the grid points. Further, we approximate the integral in the objective function with the help of the rectangle rule, confer [96]. We obtain

$$\int_0^T G\left(x\left(t\right), u\left(t\right), t\right) dt \approx \sum_{k=1}^{T} \hat{h}_k \cdot G\left(x_k, u_k, t_k\right).$$

In summary the discrete version of Problem 4.5 is:

Problem 4.16 (Discrete Optimal Control Problem)
Find coefficients x_k, v^i and u_k, $k = 1, \ldots, T$ on a given mesh $\mathbb{M}$, which solve the optimization problem

$$\max_{x_k, u_k, v^k} \sum_{k=1}^{T+1} \hat{h}_i \, \mathrm{e}^{-i_{no} t_k} \, G\left(x_k, u_k, t_k\right) - \sum_{k=1}^{T} \mathrm{e}^{-i_{no} t_k} \, G_I\left(x_{k-1}, v^k, t_k\right)$$

under the difference equations constraint

$$x_{k+1} = x_k + \hat{h}_k g\left(t_k, x_k, u_k\right),\ k = 0, \ldots, T,$$

the boundary condition

$$x_0 = \hat{x}_0,$$

the jump dynamics

$$x_k - x_{k-1} = g_I\left(x_{k-1}, v^k, t_{k-1}\right), k = 1, \ldots, T,$$

the mixed state and control constraints

$$\Phi\left(t_k, x_k, u_k\right) \leq 0,\ k = 1, \ldots, T-1,$$

the state constraints

$$\Phi_x\left(t_k, x_k\right) \leq 0,\ k = 1, \ldots, T,$$

and the control constraints

$$\Phi_u\left(t_k, u_k\right) \leq 0,\ k = 1, \ldots, T.$$

As we see Problem 4.16 is a finite dimensional, nonlinear optimization problem which can be written in the general form:

$$\begin{aligned} &\max_z J\left(z\right) \\ &\text{subject to } \tilde{G}\left(z\right) \leq 0, \\ &\tilde{H}\left(z\right) = 0 \end{aligned}$$

with the optimization variable

$$z := \left(x_0, \ldots, x_{Nn_x+n_x}, u_0, \ldots, u_{Nn_u+n_u}, v_0, \ldots, v_{Nn_v+n_v}\right),$$

the inequality constraints

$$\tilde{G}\left(z\right) := \begin{pmatrix} \Phi\left(t_0, x_0, u_0\right) \\ \vdots \\ \Phi\left(t_N, x_N, u_N\right) \\ \Phi_x\left(t_0, x_0\right) \\ \vdots \\ \Phi_x\left(t_N, x_N\right) \\ \vdots \\ \Phi_u\left(t_0, u_0\right) \\ \vdots \\ \Phi_u\left(t_N, u_N\right) \end{pmatrix}$$

and the equality constraints

$$\tilde{H}\left(z\right) := \begin{pmatrix} x_1 - x_0 h_0 - g\left(t_0, x_0, u_0\right) \\ \vdots \\ x_N - x_{N-1} h_{N-1} - g\left(t_{N-1}, x_{N-1}, u_{N-1}\right) \\ x_0 - \hat{x}_0 \\ x_1 - x_0 = g_{\mathrm{I}}\left(x_0, v^1, t_0\right) \\ \vdots \\ x_N - x_{N-1} = g_{\mathrm{I}}\left(x_{N-1}, v^N, t_{N-1}\right) \end{pmatrix}$$

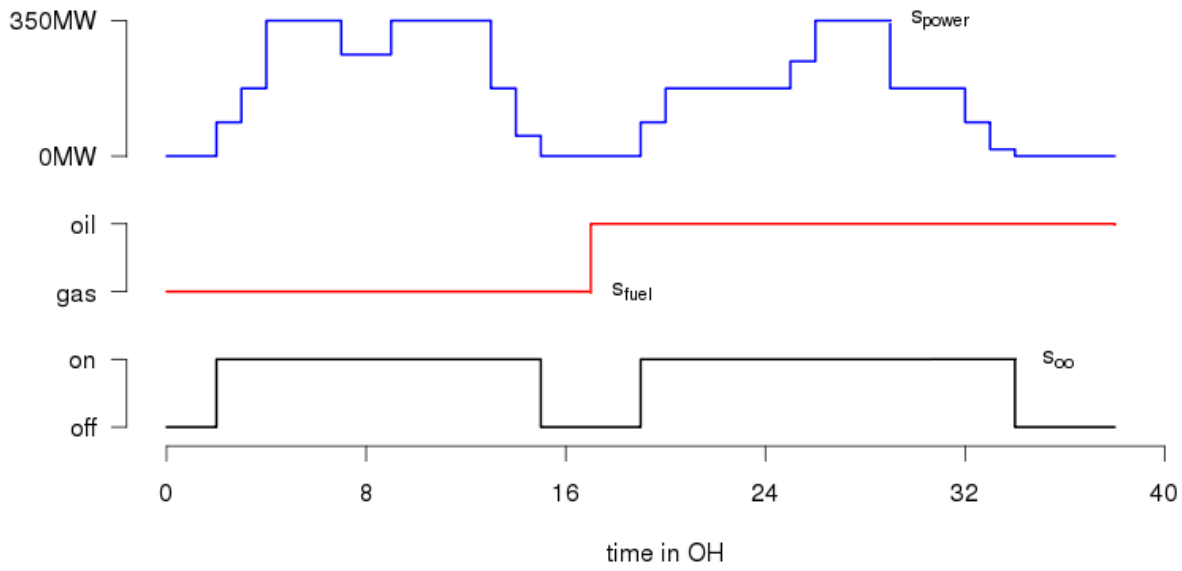


Figure 4.4.: Example of factors which influence the life consumption of a component.

We see that our problem can be very large, because we have $(n_x + n_u + n_v) \cdot (N + 1)$ optimization variables and $(n_{cx} + n_{cu}) \cdot (N + 1) + 2n_x \cdot (N + 1) + 1$ constraints.

4.3. General Maintenance Modeling

To model our replacement problem we have to introduce basic definitions and ideas which will be as well important in the later Chapters 5 and 6. We start with the definition of the life counter function related to our consideration about lifetime consumption in Section 2.2.

4.3.1. General Defintions

Life Counter

According to Chapter 2, the life consumption of a part or component depends on various factors like for example the load of the gas turbine, ambient temperature or the quality of the fuel. Therefore, the life counter must consider these effects properly. For example the life consumption of turbine blades will accelerate, if the fuel has bad quality. See Figure 4.4 for example where we see how influence parameter change over the time. Thus we could define a life counter as function of the various factors which influence the life consumption. The influencing factor changes over time and therefore the speed of life consumption will change over time. But the life counter is a monotone increasing

function. Due to these facts we define a life counter (function) in general as

$$c\colon \mathbb{R}_+ \mapsto \mathbb{R}_+, \quad t \mapsto c\,(t) \tag{4.25}$$

and we neglect the direct dependence of the influencing factor for convenience. We assume, the influencing factors are included implicitly. Therefore, the life counter masks the information how the operating factors change over time. But the life counter function gives us explicit the information of how much life is consumed. Further, the life counter $c\,(t)$ need not be continuous, like we see in our EOH-model in Section 2.2 or in particular in Figure 2.5.

Financial Data

Another important aspect is the financial data modeling, because our objective will be to maximize expected gas turbine operator's revenue. In our case we model the financial data through four main functions:

- Revenue:
$$I\colon \mathbb{R}_+ \mapsto \mathbb{R}, \quad t \mapsto I\,(t) \tag{4.26}$$
is the function which gives us information how big the revenue of the gas turbine operator is. It is implicitly connected to the operating scenario of the gas turbine, because the revenue will be lower, if the gas turbine does not run at full load or the revenue changes, if the fuel changes.

- Equipment costs:
$$C_{\mathrm{ME}}\colon \mathbb{R}_+ \mapsto \mathbb{R}, \quad t \mapsto C_{\mathrm{ME}}\,(t) \tag{4.27}$$
is the function which gives the actual cost for the parts which are replaced or more accurately to set the component back into an as new condition, e.g. by repair. That can be for example costs for spare parts or nondestructive testing.

- Assembly costs:
$$C_{\mathrm{MA}}\colon \mathbb{R}_+ \mapsto \mathbb{R}, \quad t \mapsto C_{\mathrm{MA}}\,(t) \tag{4.28}$$
is the function which gives the actual cost to dismantle a part. Further, the assembly costs reflect the outage duration, because they include the revenue loss of the gas turbine operator due to the downtime.

- Failure costs:
$$C_{\mathrm{R}}\colon \mathbb{R}_+ \mapsto \mathbb{R}, \quad t \mapsto C_{\mathrm{R}}\,(t) \tag{4.29}$$
is the function which gives the penalty fee of the gas turbine operator, if a component of the gas turbine fails.

The overall maintenance cost C_{M} is always the sum of the equipment cost C_{ME} and assembly cost C_{MA} defined by

$$C_{\mathrm{M}}\,(t) := C_{\mathrm{ME}}\,(t) + C_{\mathrm{MA}}\,(t)\,. \tag{4.30}$$

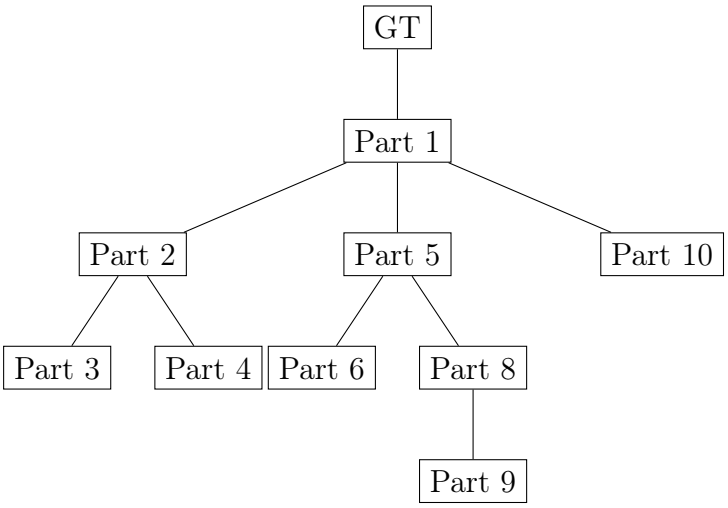


Figure 4.5.: Disassembling dependency tree $\mathcal{T}$ of a gas turbine. If we want to disassemble Part i, then we must dissemble all parts of the path between GT and Part i.

In Section 6.3 we will extend the maintenance cost definition by two additional functions $C_{\mathrm{MI}}(t)$ and $C_{\mathrm{MP}}(t)$ to model the inspection cost.

Assembly Dependencies

Due to the construction of the gas turbine there are many assembly dependencies. For example, a rotor lift requires a complete disassembly of the gas turbine. This fact must be included in our model, because we can create a benefit from it. We can save time and money, if we do several service actions at the same outage.

Therefore, we try to model the disassembly dependency with a tree $\mathcal{T} = (V, E)$ with nodes/vertices V and edges E. A schematic example is shown in Figure 4.5. The basic idea is to put the part which must be disassembled first on the top and attach parts in the order of the dismantling process. This gives us the disassembling order. Some parts may have many children and some not. Depending on the position of a part in the tree, the disassembling time will vary. A part with a deeper position in the tree needs a longer dismantle time, than a part with a higher position. The disassembling cost also increases. Every node of the tree gets the information of the additional dismantle time and assembly cost when compared to its parent. In this case, the definition of the assembly cost C_{MA} in equation (4.28) changes and gives us the additional cost, only. We get the whole disassembly time for a part, if we count all dismantle times of the path from the part node P to the tree root. Following this process, we define a function

$$T_{\mathrm{D}} \colon \mathbb{T} \times V \mapsto \mathbb{R}, \quad (\mathcal{T}, P) \mapsto T_{\mathrm{D}}(\mathcal{T}, P),$$

where $\mathbb{T}$ is the set of all possible dependency trees. If we dismantle more than one part,

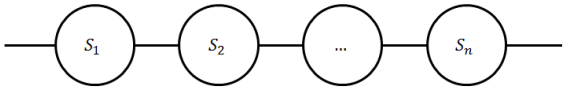


Figure 4.6.: Serial system with n parts.

e.g. n, then we get the outage duration by the maximum of the single outage durations by

$$T_{\mathrm{DO}} = \max\left(T_{\mathrm{D}}\left(\mathcal{T}, P_1\right), \ldots, T_{\mathrm{D}}\left(\mathcal{T}, P_n\right)\right)$$

If we want to replace a set of parts $\{P_1, \ldots, P_n\}$, then we get the overall disassembly cost $C_{\mathrm{MA}}(r)$ by counting every single disassembly cost $C_{\mathrm{MA}_i}(t)$ once which is need to replace every part $P_i \in \{P_1, \ldots, P_n\}$. This process is defined by

$$C_{\mathrm{MA}} \colon \mathbb{R}_+ \times \mathbb{T} \times \mathcal{P}(V) \mapsto \mathbb{R}, \quad (t, \mathcal{T}, P) \mapsto T_{\mathrm{D}}(t, \mathcal{T}, P), \tag{4.31}$$

where $\mathcal{P}(V) := \{A | A \subseteq V\}$ is the power set of V.

4.3.2. System Reliability Modeling

We extend our modeling approach by reliability modeling. Therefore, we recall our reliability functions

- Hazard rate function: $h(t)$ (Equation (2.5))
- Survival/Reliability function: $S(t)$ (Equation (2.7))
- Failure function: $F(t) = 1 - S(t) = 1 - \exp\left(-\int_0^t h(\tau)\, d\tau\right)$ (Equation (2.3))

from Section 2.1.3 and show how to connect them. But first we define that the hazard rate depends on the life counter $c(t)$, e.g.

$$h_{\mathrm{new}}(t) := h_{\mathrm{old}}(c(t)). \tag{4.32}$$

Due to the relationships in equation (2.7) the failure and survival functions $F(t)$ and $S(t)$ depend on the life counter $c(t)$. As mentioned before a gas turbine consists of multiple components and therefore system reliability is an important part of modeling. We introduce three basic rules of system reliability and show how we can use them to model more complex ones. We follow the ideas from [34] and [64].

Serial Systems

We start with the reliability of a serial system. We assume our system consists of n parts, see Figure 4.6 for an illustration. The system fails, if one part fails. Furthermore,

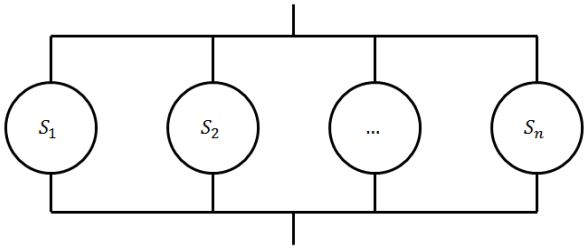


Figure 4.7.: Parallel system with n parts.

we assume that the random failure events are independent. For such a system we can calculate the reliability / survival probability by

$$S_{\text{ser}} = \prod_{i=1}^{n} S_i. \tag{4.33}$$

The reliability decreases, if the number of parts in the serial connection increases. In terms of hazard rate functions we calculate the reliability by

$$S_{\text{ser}}(t) = \exp\left(-\int_0^t \sum_{i=1}^{n} h_i(\tau)\right). \tag{4.34}$$

Parallel Systems

The second system is a parallel system which fails, if all parts fail. The schematic diagram is presented in Figure 4.7. We calculate the reliability by the complementary approach and we obtain

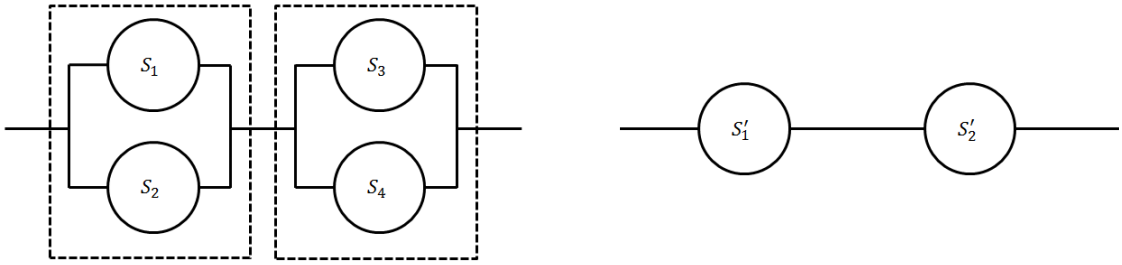
$$S_{\text{par}} = 1 - \prod_{i=1}^{n} (1 - S_i). \tag{4.35}$$

The reliability increases, if the number of parts increases.

k out of n Systems

A k out of n System consists of n parts and it works, if k or more parts did not fail. It is a mixture between a serial and parallel system. A 1 out of n System is full a parallel system and a n out of n System is a serial system. If S_i is the reliability of a single part, than we get overall reliability of a k out of n System by

$$S_{kn} = \sum_{\substack{J \subseteq N \\ |J| \geq k}} \prod_{i \in J} S_i \prod_{j \in N \setminus J} (1 - S_j) \quad \text{with} \quad N = \{1, \ldots, n\}.$$



(a) Complex system divided into subsystems by dashed rectangles.

(b) Reduced system to serial system

Figure 4.8.: Calculation method for complex system.

Complex Systems

The reliability of complex system can be calculated, if we split the system in small parallel or serial systems and calculate the reliability for the small systems. We repeat this method until we have the whole system reliability. We illustrate this method for the example in Figure 4.8(a). First at all, we divide the system 1 in two parallel systems as we see in Figure 4.8(a) marked by the dashed rectangles. For the two subsystems we get the following system reliabilities

$$S_1' = 1 - \prod_{i=1}^{2} (1 - S_i) \quad \text{and} \quad S_2' = 1 - \prod_{i=3}^{4} (1 - S_i)$$

We replace the parallel systems with the system reliabilities S_1' and S_2' and get the serial system 2 as presented in Figure 4.8(b). The new system has the reliability

$$S = \prod_{i=1}^{2} S_i' = \prod_{i=1}^{2} \left(1 - \prod_{j=1+2(1-i)}^{2i} (1 - S_j) \right).$$

This methodology give us the possibility to calculate all necessary system reliabilities for all cases in our modeling approach.

Replacement Modeling

Now, we introduce the effect of service in our replacement model. We assume that every service action resets the consumed life of a part back to zero or in other words in an as new state. Therefore, we say that all our service actions are replacements. Further, we can assume that our life counter $c(t)$ of a component is set back to zero and thus the the hazard rate is set back to zero. In general, we can redefine a life counter $c(t)$ after an replacement at time $t^\star$ by

$$c(t) := c(t) - c(t^\star), \quad t \geq t^\star. \tag{4.36}$$

4.4. The Replacement Model

In this section we show how we connect to the impulse control theory from Section 4.1.2, the modeling approach from Section 4.3 and direct solution method from Section 4.2 to solve our replacement model.

In our model we assume that the gas turbine has m different life counters $c_i(t)$ which belong to different parts or components. First, we show how we can model the effect of replacement with the help of impulse control. As mentioned in equation (4.36) after a replacement of a part, the counter is reseted to zero. We use the state variables $x_i(t)$, $i = 1, \ldots, n$ from the impulse control problem to model the resetting of the counter. We redefine the counter by

$$c_i^s(t, x_i) = c_i(t) - x_i(t), \quad i = 1, \ldots, m.$$

The new life counter fulfills

$$c_i^s(\tau_j, x_i) = 0 \quad \text{for all } j = 1, \ldots, N, \tag{4.37}$$

at every service time point τ_j for part i which was demanded. To reach this aim, we need to establish certain restrictions and dynamics for the state variables x_i, $i = 1, \ldots, m$ to consider the full resetting of the counter at the discrete time points. The following state dynamics and restrictions deliver the right behavior of the state variable in our impulse control problem:

- State dynamics:

$$\begin{aligned} \dot{x}_i(t) &= 0 \\ x_i(0) &= 0, \end{aligned} \tag{4.38}$$

- Jump dynamics:

$$x_i\left(\tau_j^+\right) - x_i\left(\tau_j^-\right) = v_i^j \left[c_i(\tau_j) - x_i\left(\tau_j^-\right)\right], \quad i = 1, \ldots, m,\ j = 1, \ldots, N, \tag{4.39}$$

- Jump control 1 for equipment:

$$0 \leq v_i^j \leq 1 \quad \text{for } i = 1, \ldots, m,\ j = 1, \ldots, N, \tag{4.40}$$

- Jump control 2 for assembly:

$$0 \leq w_i^j \leq 1 \quad \text{for } i = 1, \ldots, m,\ j = 1, \ldots, N, \tag{4.41}$$

- Disassemble dependencies:

$$v_i^j \leq w_k^j \quad \text{for } i = 1, \ldots, m,\ k \in P_i,\ j = 1, \ldots, N, \tag{4.42}$$

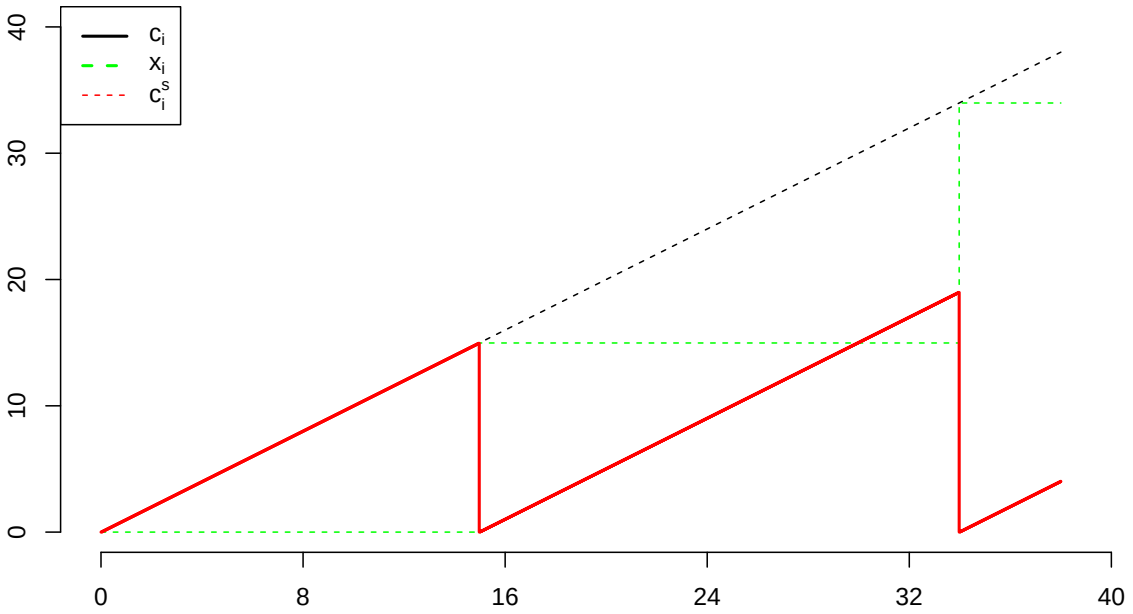


Figure 4.9.: Service state modeling: Dashed black line is the counter $c_i(t)$, the dashed green line $x_i(t)$, the bold red line $c_i^s(t, x_i)$.

- State restrictions:

$$0 \leq x_i(t) \leq c_i(t) \quad \text{for } t \in [0, T] \text{ and } i = 1, \ldots, m, \tag{4.43}$$

where τ_j, $j = 1, \ldots, N$, are the jump times and give us the service time points. Condition 4.38 guarantees that our life counter c_i^s is manipulated by service only at service times. The condition 4.39 delivers the right "jump height" for our state variables x_i, $i = 1, \ldots, m$.

Further, it guarantees that we get $v_i^j = 1$ or $v_i^j = 0$. Strictly, the restrictions 4.40 and 4.41 should be

$$v_i^j, w_i^j \in \{0, 1\} \quad \text{for } i = 1, \ldots, m \quad \text{and} \quad j = 1, \ldots, N.$$

Since our modeling approach meets the conditions for a bang-bang solution, we can use the relaxed version from equations (4.40) and (4.41) and we see the linear dependency of the control variables v_j^j and w_i^j in equation (4.39). We will derive the linear dependency of the objective function later in equation (4.44).

Equations (4.40) and (4.41) give us the information to change part i and disassemble part k at time τ_j. In condition 4.42 the set $P_i \subseteq \{1, \ldots, m\}$ gives us the disassemble dependencies. More in detail, P_i is the set of all parts which must be disassembled when we want to replace part i. The impulse dynamics are shown schematically in Figure 4.9.

The hazard rates with service are defined by

$$h_i^s(t) = h_i\left(c_i^s(t, x_i)\right), \quad i = 1, \ldots, m.$$

The definition is equal to our old version. We only changed the life counter. We model our multi component system as a serial connection according to section 4.3.2 and therefore, according to equation (4.34), we obtain for the overall hazard rate

$$h^{\mathrm{s}}(t) := \sum_{i=1}^{m} h_i^{\mathrm{s}}(t) = \sum_{i=1}^{m} h_i\left(c_i^{\mathrm{s}}(t, x_i)\right)$$

and for the overall survival function

$$S(t) := \exp\left(-\int_0^t \sum_{i=1}^{m} h_i^{\mathrm{s}}(\tau)\, d\tau\right) = \exp\left(-\int_0^t \sum_{i=1}^{m} h_i\left(c_i^{\mathrm{s}}(t, x_i)\right) d\tau\right).$$

Now, we are able to define the cost for the j-th jump:

$$G_{\mathrm{I}}\left(x\left(\tau_j^-\right), v^j, w^j, \tau_j^-\right) = \mathrm{e}^{-i_{\mathrm{no}}\tau_j^-}\, S\left(\tau_j^-\right) \sum_{i=1}^{m} \left(C_{\mathrm{ME}i}\left(\tau_j^-\right) v_i^j + C_{\mathrm{MA}i}\left(\tau_j^-\right) w_i^j\right). \quad (4.44)$$

and it gives us the expected cost for service in present value formulation. Thereby $i_{\mathrm{no}} \in [0, 1]$ is the discount factor. Equation (4.44) and the restriction in equation (4.42) represent the assembly cost C_{MA} in Equation (4.42).

The continuous cost function G is given by

$$\begin{aligned} G(x(t), u(t), t) &= \mathrm{e}^{-i_{\mathrm{no}}t}\, S(t) \left[I(t) - C_{\mathrm{R}}(t) \sum_{i=1}^{m} h_i^{\mathrm{s}}(t)\right] \\ &= \mathrm{e}^{-i_{\mathrm{no}}t}\, S(t) \left[I(t) - C_{\mathrm{R}}(t)\ h^{\mathrm{s}}(t)\right] \end{aligned}$$

and it handles the expected revenue and risk cost. We remark that an extension to different C_{R_i} is possible to take different risks into account. In summary, our model problem with replacement is given by:

Problem 4.17 (Replacement Model)
Find a number N of outage times, time points $\tau \in \mathbb{R}_+^N$, $\tau_j < \tau_{j+1}$, replacement control $v^j \in \mathbb{R}^m$, $j = 1, \ldots, N$ and disassemble control $w \in \mathbb{R}^m$, $j = 1, \ldots, N$ which solves

$$\begin{aligned} \max_{N, v^j, w^j, \tau} \int_0^T & \mathrm{e}^{-i_{no}t}\, S(t) \left(I(t) - C_R(t)\, h_i^s(t)\right) dt \\ & - \sum_{j=1}^{N} \mathrm{e}^{-i_{no}\tau_j^-}\, S\left(\tau_j^-\right) \sum_{i=1}^{m} C_{MEi}\left(\tau_j^-\right) v_i^j + C_{MAi}\left(\tau_j^-\right) w_i^j \end{aligned} \quad (4.45)$$

subject to the constraints in equations (4.38) *to* (4.43).

Now, we need to solve Problem 4.17 to get our optimal maintenance schedule. We use the well established direct solution method from Section 4.2. We also use all discretizing steps as mentioned in Section 4.2 with an equidistant mesh $\mathbb{M} = \{t_1, \ldots, t_T\}$ with step size $\hat{h}$. In summary, we get the following discrete version of Problem 4.17 which is our starting point for the numerical analysis in the next Section 4.5:

Problem 4.18 (Discrete Replacement Model)
Find state vectors $x_i \in \mathbb{R}^T$ and control vectors $v^i, w^i \in [0,1]^T$, $i = 1, \ldots, n$ which solves

$$\max_{x_i, v^i, w^i} \hat{h} \sum_{k=1}^{T} \mathrm{e}^{-i_{no} t_k} S(t_k) \left[I(t_k) - C_R(t_k) h_i^s(t_k) \right] - \sum_{k=1}^{T} \mathrm{e}^{-i_{no} t_{k-1}} S(t_{k-1}) \sum_{i=1}^{m} C_{MEi}(t_{k-1}) v_i^k + C_{MAi}(t_{k-1}) w_i^k \tag{4.46}$$

subject to the discrete jump dynamics

$$x_i(t_k) - x_i(t_{k-1}) = v_i^k \left[c_i(t_k) - x_i(t_{k-1}) \right], \ i = 1, \ldots, m, \ k = 1, \ldots, T,$$

the jump control for equipment

$$0 \leq v_i^k \leq 1, \quad i = 1, \ldots, m, \ k = 1, \ldots, T,$$

the jump control for assembly

$$0 \leq w_i^k \leq 1, \quad i = 1, \ldots, m, \ k = 1, \ldots, T,$$

the disassemble dependencies,

$$v_i^k \leq w_l^k, \quad i = 1, \ldots, m, \ l \in P_i, \ k = 1, \ldots, T,$$

and the state restrictions

$$0 \leq x_i(t_k) \leq c_i(t_k), \ i = 1, \ldots, m, \ k = 1, \ldots, T,$$

to get the optimal outage schedule.

We can read off the set of optimal service time points $T^\star := \{t_1^\star, \ldots, t_N^\star\}$ from the optimal solution of Problem 4.18 by searching for all $k = 1, \ldots, T$ with $v_i^k = 1$, $i = 1, \ldots, m$ and set $t_k = t_{k_l}^\star$, $l = 1, \ldots, N$. Further, v_i^k encodes the replacement of the i-th part at time t_k.

4.5. Numerical Analysis

In this Section we present numerical results from our discrete replacement model. We start with a description of the numerical implementation and then present different results concerning mesh sizes. various model sizes, input data and comparison between actual maintenance concepts and the optimized one.

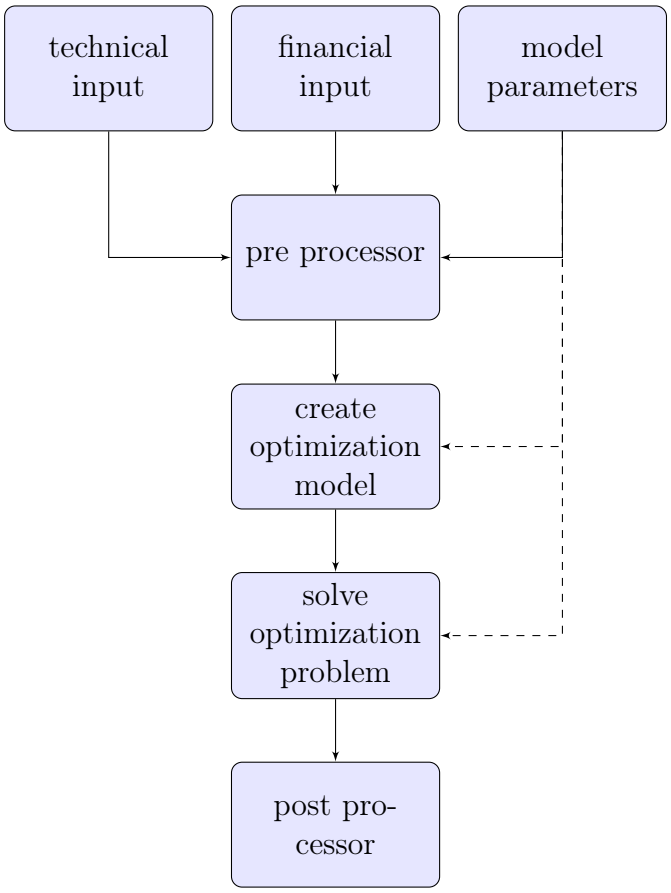


Figure 4.10.: Flow chart of the replacement model implementation.

4.5.1. Numerical Implementation

The software to solve Problem 4.18 was written in R 3.2.3 which is programming language and software environment for statistical computing, confer [75]. Figure 4.10 illustrates the general structure of the implementation. First, we have to define the various model parameters which we divide into three groups: The first group contains all technical data like the number of components, the failure probabilities, assembly dependencies and the life counter functions. The second group includes all financial data like the revenue, the risk cost, service cost and the time duration. The last group takes model parameters into account which we need for the discretization and the control of the nonlinear solver.

The preprocessor block takes financial and technical input and adapts them to the model parameters. This step is necessary, because we need to connect the continuous given values like the gas turbine operator's revenue $I(t)$ to the discrete values $I_k = I(t_k)$ at time t_k which are influenced by the step size $\hat{h}$. We describe the process for the financial data shortly. We assume that a mesh $\mathbb{M}_{\hat{h}} = \{t_0 < \dots t_T\}$ and a revenue function $I(t)$

are given. Then we define

$$I_k := \frac{1}{t_{k+1} - t_k} \int_{t_k}^{t_{k+1}} I(t)\, dt, \quad k = 0, \ldots, T-1 \tag{4.47}$$

as mean value of the revenue I in the interval $[t_k, t_{k+1}]$. We make the same approach for the risk and maintenance cost:

$$C_{\mathrm{R}k} := \frac{1}{t_{k+1} - t_k} \int_{t_k}^{t_{k+1}} C_{\mathrm{R}}(t)\, dt, \quad k = 0, \ldots, T-1 \tag{4.48}$$

$$C_{\mathrm{ME}k} := \frac{1}{t_{k+1} - t_k} \int_{t_k}^{t_{k+1}} C_{\mathrm{ME}}(t)\, dt, \quad k = 0, \ldots, T-1 \tag{4.49}$$

$$C_{\mathrm{MA}k} := \frac{1}{t_{k+1} - t_k} \int_{t_k}^{t_{k+1}} C_{\mathrm{MA}}(t)\, dt, \quad k = 0, \ldots, T-1 \tag{4.50}$$

With this preprocessed data we are able to set up and solve the optimization problem. We use the open source nonlinear solver IPOPT 3.11.9 (**I**nterior **P**oint **OPT**imizer) which was developed by Wächter, confer [92]. IPOPT uses a interior point method to solve the optimization problem. The last block post process the result data from solver to present the solution.

4.5.2. Numerical results

We present four different case studies for validation purposes. First, we present the difference between the actual outage schedule and the optimized one. Next, we show how the solution is influenced by the size of the mesh. Then we analyze how financial data or operating regimes influence the outage schedule. As last step we increase the number of the counters to show that the problem is still solvable in a reasonable computation time.

We used for the following calculations a Linux workstation with an Intel Xeon E5-2643 v3 3.4GHz CPU and 256GB RAM.

Comparison of the Optimal vs Standard Outage Schedule

We start with an example of a gas turbine with three lifetime counters - a gas turbine which consists of three components $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$, respectively. The three components represent important component groups like the hot gas path parts, the compressor and the rotor & casings. The important quantities of the model are summarized in Table 4.1. The financial data and other model data are based on the sources in Section 3.3, but they are fictitious for all following examples in this thesis. In Figures 4.12(a) and 4.12(b) we present the financial data. The revenue of the gas turbine operator changes between high price periods and low prices periods, confer Figure 4.12(b). This time depending

Model parameter	
Number of counters:	3
Time duration:	12 years
Risk:	Complete loss of the GT (3330.0Mio. Euro)
Failure probabilities:	Weibull Model with parameters: $m_{1,2,3} = 2.12$ and $\eta_1 = 8.5 \cdot 10^5$, $\eta_2 = 1.2 \cdot 10^6$, $\eta_3 = 2.2 \cdot 10^6$
Life counter:	$c_3(t) = 2.2c_1(t) = 2.0c_2(t)$ for all t (ALL EOH-Model)
Power:	300MW
Operating hours per year:	8030
Starts per year:	30
Equipment cost:	$C_{\text{ME1}} = 9.0 \cdot 10^5$, $C_{\text{ME2}} = 1.7 \cdot 10^6$, $C_{\text{ME3}} = 7.0 \cdot 10^5$
Assembly cost:	$C_{\text{MA1}} = 0.3C_{\text{ME1}}$, $C_{\text{MA2}} = 0.3C_{\text{ME2}}$, $C_{\text{MA3}} = 0.3C_{\text{ME3}}$
Outage duration:	$C_{\text{D1}} = 10$days, $C_{\text{D2}} = 5$days, $C_{\text{D3}} = 4$days

Table 4.1.: Model parameters of the 3 counter model.

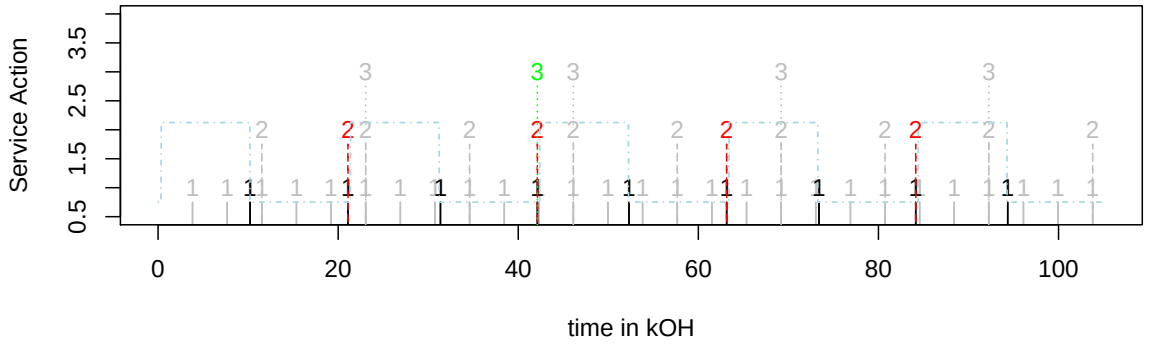


Figure 4.11.: The Figure presents the optimal outage schedule for the 3 counter problem. The gray marked schedule represents the standard schedule and the black, red and green colored represents the optimized schedule. The blue dashed line indicates the electrical price profile.

price profile is typical for various regions in the world, for example in countries where people use air condition a lot during summer. Further, the dismantle dependencies are very simple: If we want to disassemble component $\mathcal{C}_i$, then we must disassemble all components $\mathcal{C}_j$ with $j < i$. In addition, if we replace component $\mathcal{C}_i$, then we replace the components $\mathcal{C}_j$ with $j < i$, also.

In Figure 4.11, we see a comparison between an optimal and standard maintenance schedule for the three counter example. The gray marked outage schedule applies to the standard outage schedule. This standard schedule refers to the EOH model from Section 2.3. We define for that reason a new global life counter $c_{\text{g}}(\cdot)$ by

$$c_{\text{g}}(t) := \max\{c_1(t), c_2(t), c_3(t)\}$$

to reflect that we have only one life counter. In the global life counter approach the highest stressed component is always leading. According to the counter c_{g} we replace

n	$\hat{h}$ in hours	optimal value	runtime in s	iterations
200	525.60	4.8972141e+04	3.0	105
500	210.24	5.2865214e+04	9.9	158
1000	105.12	5.4111597e+04	32.5	238
2000	52.56	5.5588646e+04	76.9	231
5000	21.02	5.6338496e+04	463.2	222
7000	15.02	5.6490020e+04	982.8	364
9000	11.68	5.6480962e+04	1094.4	255
13000	8.09	5.6617032e+04	2757.8	316
17000	6.13	5.6667140e+04	4909.1	346

Table 4.2.: Convergence study, number of iterations and runtime for different mesh sizes $\hat{h}$.

every 8333EOH component 1, every 25000EOH component 2 and every 50000EOH component 3. The black, red and green colored optimal outage schedule in Figure 4.11 is less dense than the standard outage schedule and fits better to the electrical price profile. The optimal value of the objective function is 56.33Mio. Euro compared to 0.03Mio. Euro in the standard case. The revenue was increased by the factor $\frac{56.33\text{Mio}}{0.03\text{Mio}} = 1877.7$.

Influence of Mesh Size

Now, we start analyzing the effect of the mesh size on the solution to our problem. We take the same model as in the last section. We assume that the step sizes of the mesh

$$\mathbb{M}_{\hat{h}} := \{t_0 = 0 < t_1 < \ldots < t_T = T_{\text{end}}\}$$

are constant. Therefore, we obtain $\hat{h} = \frac{T_{\text{end}}}{n}$. In Table 4.2 we see how the mesh size changes the optimal solution. We identify that the optimal solution does not change significantly, if we choose $n > 5000$. The relative difference between the optimal solution for $n = 5000$ and $n = 17000$ is

$$\left|1 - \frac{5.6490020e+04}{5.6667140e+04}\right| \cdot 100\% \approx 0.313\% \tag{4.51}$$

Further, the optimal outage schedules are presented in Figure 4.12(c). As we see, the outage schedule stays quite the same, if we increase n over 5000. Therefore, we can assume that a mesh size $\hat{h} = 21.02\text{h} \approx 1\text{day}$ is fine enough. From a practical view this step size is small enough, because outages are planed on a daily basis or an even longer times basis. We can conclude that our approach reaches convergence in reasonable runtimes well below one hour.

Parameter Variation

In this section we present how the change of model parameters like financial data or failure probabilities influences the solution. This study is important to analyze the sensitivity of the solution influenced by estimated input parameter. First, we keep the high price profile from the last sections and change the failure probabilities. We change only the scale factors η_i of the Weibull distribution. We use the following sets for η_i:

$$
\begin{aligned}
E_0 &= \left\{\eta_1 = 8.50 \cdot 10^5, \eta_2 = 1.2 \cdot 10^6, \eta_3 = 2.2 \cdot 10^6\right\}, \\
E_1 &= \left\{\eta_1 = 4.25 \cdot 10^5, \eta_2 = 6.0 \cdot 10^5, \eta_3 = 1.1 \cdot 10^6\right\}, \\
E_2 &= \left\{\eta_1 = 1.70 \cdot 10^6, \eta_2 = 2.4 \cdot 10^6, \eta_3 = 4.4 \cdot 10^6\right\}, \\
E_3 &= \left\{\eta_1 = 3.40 \cdot 10^6, \eta_2 = 4.8 \cdot 10^6, \eta_3 = 8.8 \cdot 10^6\right\}.
\end{aligned}
$$

Further, we use $n = 5000$ for the time discretization. The resulting outage schedules are presented in Figure 4.13. As anticipated, the density of outages decreases, if the failure probabilities decreases which is connect to the change of the scale parameter η. But a special feature of the high price period model is that all bigger / longer or more expensive outages are scheduled before or after the high price time period. This behavior is reasonable, because it decreases the risk of a failure in a high price period and it minimize the revenue loss in the high price period.

Next, we change the revenue profile to a constant revenue profile with 3 Euro per MWh. The service cost are constant, too. The other model parameters are the same as in the first model. The resulting outage schedule is illustrated in Figure 4.14. We see that the outage schedule has a fixed pattern. Every second outage is of type 2 and the time difference between two outages is always the same, approximately 10400 hours.

As last parameter variation, we give an example where we use real market data from the "European Energy Exchange" (EEX) to show that our approach works with real revenue data. In Figure 4.15(a) we see the revenue data which give us the the price for one baseload hour in the day ahead market, but the presented data does not include the fuel cost. This data was made available for this thesis by the EEX, confer [36]. Figure 4.15(b) illustrates the associated service costs and the final outage schedule is presented in Figure 4.15(c). The outage schedule is very dense compared to our previous examples, because the revenue is higher as in the other models and therefore the impact of the penalty fee of failure is not big.

Counter Variation

In our last example we increase the number of counters and look at the calculation time in terms of time and iterations. The size of our optimization problem strongly depends on n (time discretization) and m (components) . We have $3nm$ optimization variables and $4mn - n - m$ constraints. We take our high price model from above and further

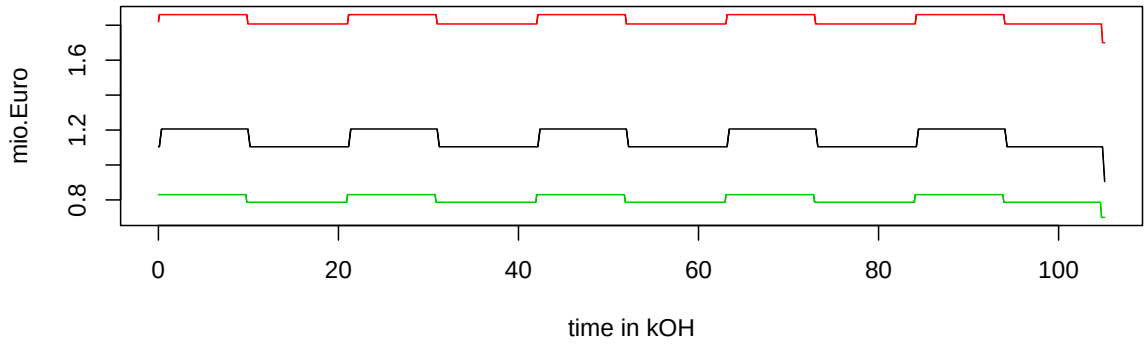
component	η_i	$C_{\mathrm{ME}i}$	$C_{\mathrm{D}i}$	c_i
4	$3.2 \cdot 10^6$	600000.0	2	1.5
5	$4.2 \cdot 10^6$	500000.0	2	1.0
6	$5.2 \cdot 10^6$	450000.0	2	1.1
7	$6.6 \cdot 10^6$	400000.0	2	1.0
8	$7.6 \cdot 10^6$	300000.0	2	1.0
9	$8.7 \cdot 10^6$	250000.0	2	1.1
10	$9.8 \cdot 10^6$	200000.0	2	1.0

Table 4.3.: The model data for the different components.

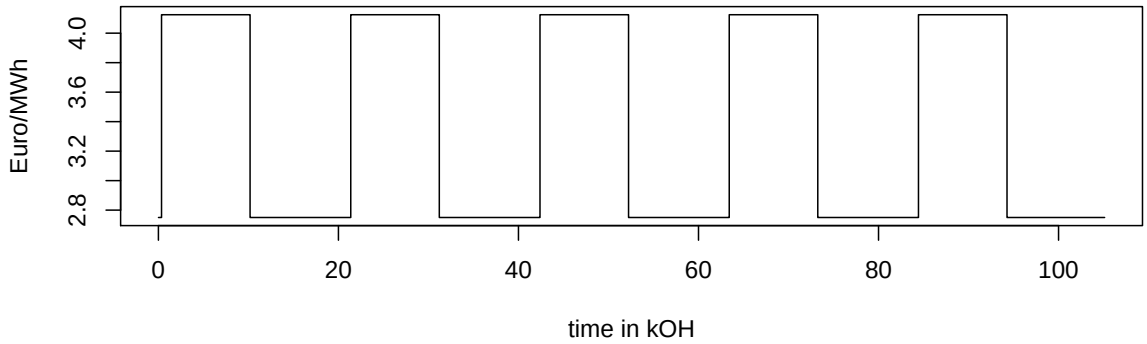
# components	iterations	runtime in sec	# variables	# constraints
3	222	463.2	45000	54997
5	358	1823.5	75000	94955
8	362	2538.7	120000	154992
10	392	4264.5	250000	194990

Table 4.4.: Runtime, iteration count and model size for different number of components.

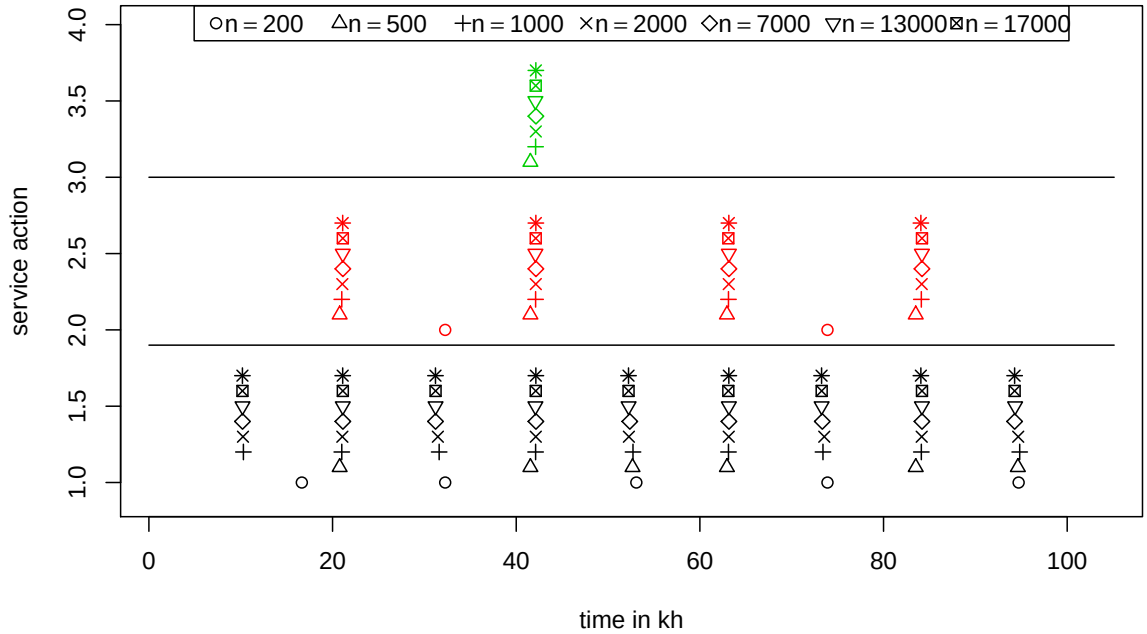
we extend it by more lifetime counters. The model data for the additional counters are presented in Table 4.3. The results are presented in Table 4.4. We can conclude that we also can solve problems with a high number of life counters, needed for real world applications like our gas turbine outage scheduling problem, in an acceptable computation time. If we take the rotor, compressor, casing, combustion chamber and the four turbine stages into account, then we need 8 lifetime counters, corresponding to less than one hour for one optimization.



(a) Maintenance cost equipment + assembly, $n = 17000$.



(b) High price period revenue trend, $n = 17000$.



(c) Optimal outage schedules for different n. The black / red / green symbols correspondent to replacement of component 1 / 2 / 3,

Figure 4.12.: Financial data and results for the three life counter examples with high price periods in the revenue.

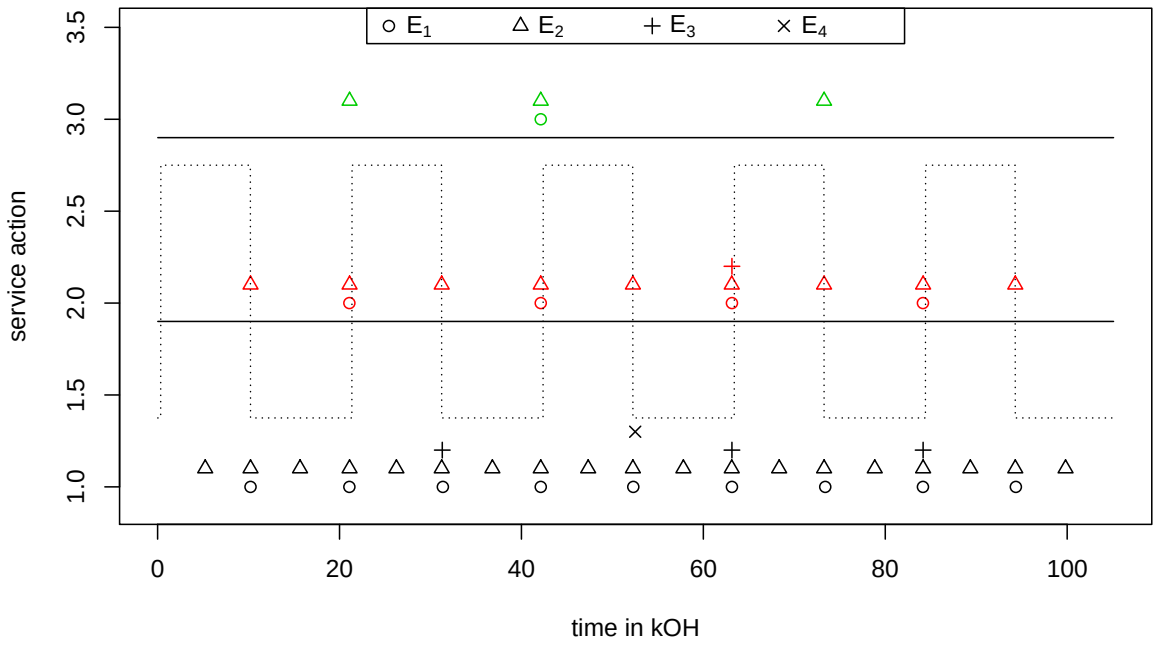


Figure 4.13.: Optimal outage schedules for different sets E_i of the Weibull shape parameter η. The dashed line shows the revenue trend to support the representation of the revenue influence to the outage time points.

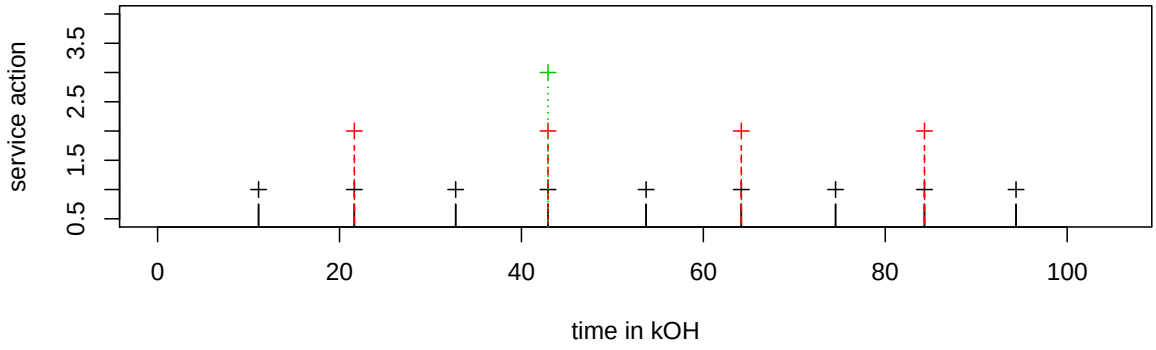
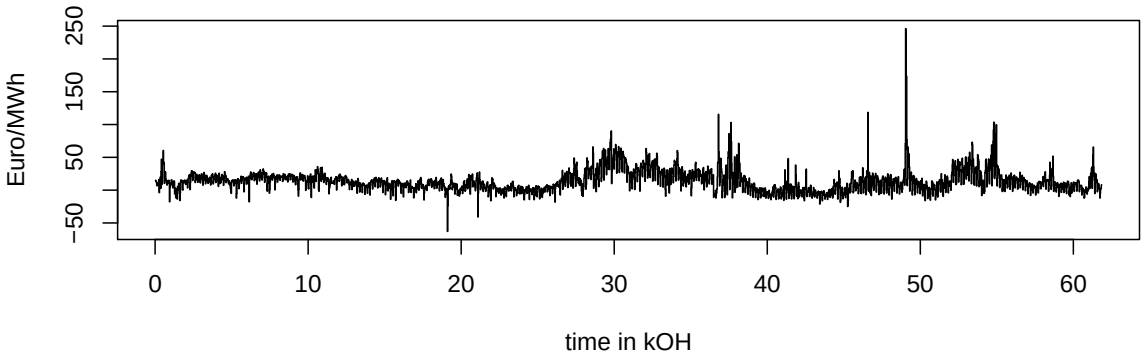
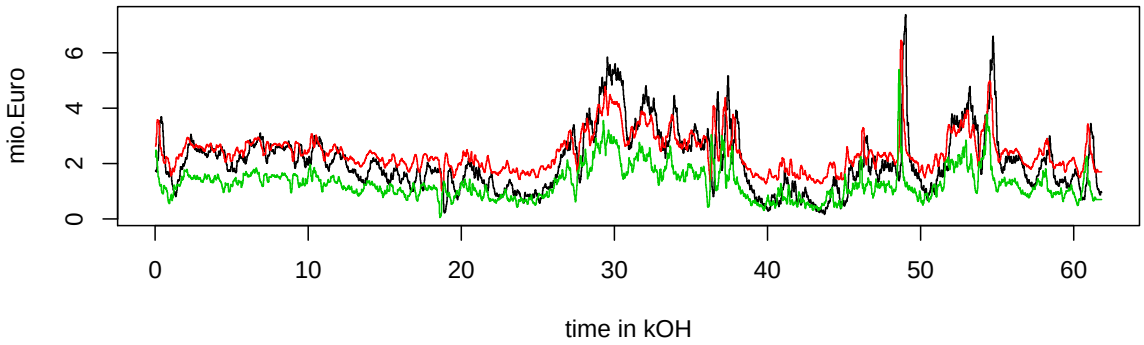


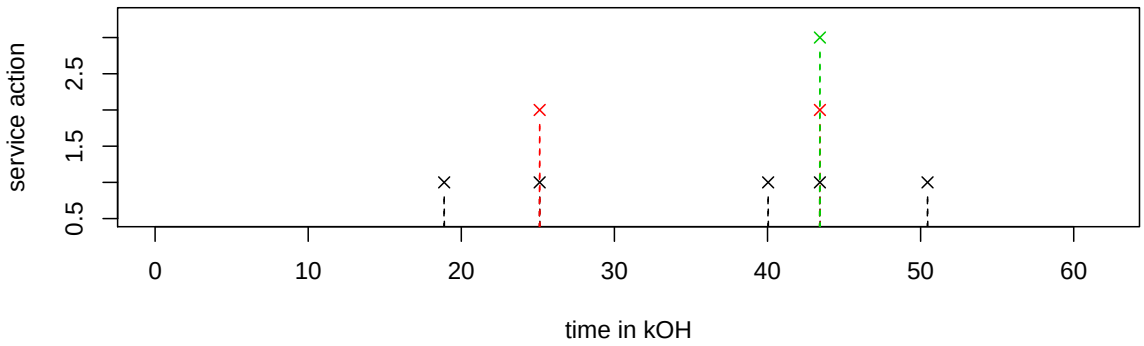
Figure 4.14.: Optimal outage schedule for constant revenue I. The black + signs belongs to service action 1, the red + signs belong to service action 2 and the green + sign belongs to service action 3.



(a) Revenue data from EEX for a daily baseload hour, confer [36].



(b) Maintenance cost according to the EEX data.



(c) Outage Schedule for EEX data. The black cross refers to service action 1, the red cross refers to service action2 and the green cross refers to service action 3.

Figure 4.15.: Financial data and results for the three counter example with EEX data.

5. Advanced Replacement Model

In the last chapter we presented a method to calculate the optimal outage schedule of a gas turbine, where the gas turbine or a particular component is not replaced after a failure. This is a valid approach, because the failure of components like the rotor will yield to a complete loss of the gas turbine. But in reality, there are component failures which won't yield to complete damage of the gas turbine like for example a failure of a turbine blade. It is possible to repair the gas turbine after turbine failure in a relatively short time period and go back into operation. Mostly, we find this situation in real applications of gas turbine maintenance. Therefore, we would like to establish a model which includes this feature. Further, we have to figure out, if potentially the replacement of additional components during a forced outage event is beneficial.

Therefore, we need to find an optimal maintenance strategy which depends on the history of our gas turbine and the future operating regime of gas turbine including possible future failures. The new solution approach will deliver an optimal maintenance policy depending on the actual time step and state.

We establish the mathematical framework of Dynamic Programming to model this new feature. Therefore, we present the basic theory of Markov Decision Processes and Dynamic Programming. Then we describe our replacement model in terms of Dynamic Programming and present a numerical analysis. Finally, we show an approximate solution for the advanced replacement model with numerical results.

The usage of an approximate method is necessary, because the Dynamic Programming approach suffers from the curse of dimensionality. The model grows in the number of components exponentially and therefore exact solution methods become unattractive. The approximate methods can mitigate this effect.

The work of Bellman and Howard, confer [8] and [50], made the theory of Dynamic Programming popular. But Cayley presented in 1875 first results about stochastic sequential decision making, confer [20]. Modern research about sequential decision making started after 1940 with the work of Wald and Massé, confer [59] and [93]. Both authors presented fundamental theory insight of the decision problem and Wald presented an analysis of water resource management problems. Further analyzed problems are stochastic inventory problems, confer [4], pursuit problems, confer [52], and aircraft engine maintenance problems, confer [2]. An important text book about approximate Dynamic Programming was published by Powell in 2007, confer [73]. The presented methods were used to solve an energy dispatch problem, confer [66]. This short his-

torical overview followed the informations of [74]. As compared to the maintenance problem from [2] we use more components and we present an approximate solution methods which exploit the structure of the problem.

5.1. Markov Decision Theory

We start with a brief introduction to Markov Decision Process (MDP) theory. In this section we essentially follow [74]. But first, we start with the definition of stochastic processes, confer [69], because the Markov Decision Theory underlies a Markov process.

Definition 5.1 (Stochastic Process)
Given a probability space $(\Omega, \mathcal{F}, P)$ and a measurable space (S, Σ), a S-valued stochastic process X is a collection $\{X_t : t \in T\}$ of S-valued random variables X_t on Ω, indexed by a totally ordered set T. In particular for every fixed t, X_t is a random variable. The space S is then called the state space of the process.

In general a stochastic process can be time discrete or time continuous. Here, we consider only time discrete processes. We consider only a set of finite or countable states. In a stochastic process the system state changes at every time step according to a probability distribution. The transition probabilities can be presented as a matrix. For example, we have a state space $S = \{1, 2, 3\}$ and we get the following matrix notation for the transition probabilities:

$$\begin{array}{c|ccc} & \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \hline \mathbf{1} & 0.2 & 0.5 & 0.3 \\ \mathbf{2} & 0.6 & 0.2 & 0.2 \\ \mathbf{3} & 0.1 & 0.2 & 0.7 \end{array} \tag{5.1}$$

The matrix entry p_{ij} gives us the probability to change from state i to state j. We call such a matrix $\mathbf{P}$ stochastic matrix if:

1. $p_{ij} \geq 0$ for all i, j,
2. $\sum_j p_{ij} = 1$ for every i.

We can interpret every row of $\mathbf{P}$ as a discrete probability distribution and these are conditional probabilities. For more information see Appendix A. The condition is given by the actual state. With this observation we can define a Markov chain which is a special case of a stochastic process. We need a countable set $\mathcal{S}$, the state space, a distribution F on $\mathcal{S}$ and a stochastic matrix $\mathbf{P} = [p_{ij}]_{i,j=1}^{n}$. A sequence $\{X_t\}$, $X_t : \Omega \mapsto \mathcal{S}$, $t \in \mathbb{N}_0$, of random variables with values in $\mathcal{S}$ is called $(p_0, \mathbf{P})$-Markov-Chain, if

1. $Pr(X_0 = i) = Pr_0(i) = p_0(i)$ for all i and

2. For every $t \in \mathbb{N}_0$ and $j \in \mathcal{S}$ and for all $(n+1)$-tuples $(i_0, \ldots, i_t) \in \mathcal{S}_{t+1}$ with $Pr(X_0 = i_0, \ldots, X_t = i_t) > 0$ it holds

$$Pr(X_{t+1} = j | X_0 = i_0, \ldots, X_t = i_t) = Pr(X_{t+1} = j | X_t = i_t) = p_{i_n j}.$$

The first condition is called initial distribution. The second one is called Markov property and mean that the transition probability does not depend on the history of the process. It depends only on the actual state and time. Also, we can define a Markov chain on finite time horizon T, e.g. $0 \leq t \leq$T. In Markov chains there can exists a special class of states. We label them with e. They have the properties $p_{ee} = 1$ $p_{ej} = 0$ for all $j \neq e$. They are called terminal states, because in this state the process stops or more exactly the state does not change any more.

Next, we show how to calculate multi step transition probabilities like the following

$$Pr(X_2 = 1 | X_0 = 3), \tag{5.2}$$

where we take the stochastic matrix from equation (5.1) into account. For the mentioned example in equation (5.2) we obtain

$$\begin{aligned} Pr(X_2 = 1 | X_0 = 3) &= \sum_{k=1}^{3} Pr(X_2 = 1, X_1 = k | X_0 = 3) \\ &= \sum_{k=1}^{3} Pr(X_2 = 1 | X_1 = k, X_0 = 1) \cdot Pr(X_1 = k | X_0 = 3) \\ &= \sum_{k=1}^{3} Pr(1|k) \, Pr(k|3) \\ &= \sum_{k=1}^{3} p_{k1} p_{3k}. \end{aligned}$$

The last term is the same as $(3,1)$-th entry of the matrix $\mathbf{P}^2$. By induction on n it is easy to see that

$$Pr(X_{m+n} = j | X_m = i) \tag{5.3}$$

is the same as the (i,j)-th entry of the matrix $\mathbf{P}^n$ which was proved by Chapman and Kolmogorow, confer [55]. This result will be important in a later section of the thesis by a small modification which we introduce now.

A further important result is to know the absolute probability to be in state i and how it is influenced by the initial distribution Pr_0. We get

$$Pr(X_t = j) = \sum_{i=1}^{n} Pr(X_0 = i) \, Pr(X_m = j | X_0 = i) = \sum_{i=1}^{n} Pr_0(i) \, Pr(j|i) \tag{5.4}$$

where Pr_0 is the initial distribution of the $(Pr_0, \mathbf{P})$-Markov Chain.

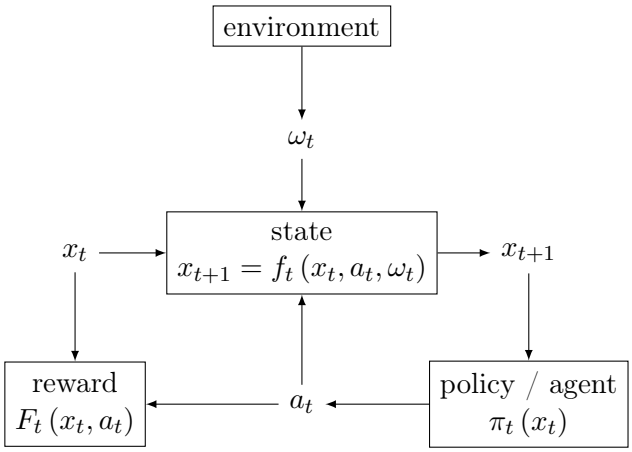


Figure 5.1.: Schematically representation of the MDP and Dynamic Programming Framework.

We expand a Markov Chain to a Markov Decision Process (MDP) by introducing a control a. In particular the transition probability changes to

$$Pr(X_{t+1} = i | X_t = j, a) =: Pr(i|j, a)$$

and thus our stochastic matrix $\mathbf{P}$ depends on the control variable a, e.g. $\mathbf{P}(a)$. With the action a we influence our stochastic matrix $\mathbf{P}$. In summary we obtain the definition from [29]:

Definition 5.2 (Markov Decision Process)
A Markov Decision Process (MDP) consists of a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, Pr, Pr_0, r, i_{no}, T \rangle$ *where*

- $\mathcal{S} \subset \mathbb{R}^d$, $d \in \mathbb{N}$, *is a finite set of discrete states* x,
- $\mathcal{A} \subset \mathbb{R}^m$, $m \in \mathbb{N}$, *is a finite set of actions* a,
- $Pr \colon \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, 1]$ *is a transition function which denotes the conditional probability*

  $$Pr(x_{t+1}|x_t, a) := Pr(X_{t+1} = x_{t+1}|X_t = x_t, a)$$

  to change from state j *under action* a *to state* i *which fulfills the Markov property. Further, it holds* $\sum_i p_{ji}(a) = \sum_i Pr(i|j, a) = 1$ *for all* j *and* a, *e.g.* $\mathbf{P}(a)$ *is a stochastic matrix for every* a,
- p_0 *is the initial states distribution at* $t = 0$,
- $r \colon \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ *is a reward function which denotes the received reward when we execute action* a *in* x.
- α *is the discount factor and*

- *T is finite time horizon.*

A Markov Decision Process and a Markov Chain are called stationary, if the transition probability does not depend on the actual stage $t = 1, \ldots, T$. Further, we can calculate the probability $Pr\left(X_{t+n} = j | X_t = i\right)$ by the matrix product $\mathbf{P}\left(a_1\right) \cdot \ldots \cdot \mathbf{P}\left(a_n\right)$ for a given action sequence $a_1, \ldots, a_n$. This result is based on the work of Chapman and Kolmogorow about the multi step transition from equation (5.3).

The MDP is the connection to Dynamic Programming approach, because it describes how the state of our model changes over the time. The general framework is illustrated in Figure 5.1.

In general the state $x \in \mathcal{S}$ of a system can consist of n subsystem with substates x_i, $i = 1, \ldots, n$. In particular, we have $x = (x_1, \ldots, x_n) \in \mathcal{S}_1 \times \ldots \times \mathcal{S}_n$. The action a can be also divided into n subactions a_i for every subsystem x_i. For this case our conditional probability to change from state x to x' under action a can be presented in general as

$$Pr\left(x' | x, a\right) = Pr\left(x'_1, \ldots x'_n | x_1 \ldots, x_n, a_1, \ldots, a_n\right).$$

We call our MDP transition independent, if it satisfies

$$Pr\left(x' | x, a\right) = Pr\left(x'_1, \ldots x'_n | x_1 \ldots, x_n, a_1, \ldots, a_n\right) = \prod_{i=1}^{n} Pr\left(x'_i | x_i, a_i\right).$$

In this case the n subsystems are independent of each other. The actions for one subsystem do not influence the behavior of the other systems. If our reward function r can be represented as

$$r\left(x, a\right) = \sum_{i=1}^{n} r_i\left(x_i, a_i\right),$$

then our MDP decomposes into n single MPD's where one agent controls one subsystem. In this case we can solve every MDP independently to get our solution. In general there is an advantage, if the agents carry out different actions at the same time. We establish a joint reward function $r_j\left(a_1, \ldots, a_n\right)$ which represents the gained reward by executing the joint action $a = (a_1, \ldots, a_n)$. A joint reward function r_j arises for example in maintenance scheduling for gas turbines. Due to the construction of the gas turbine, we can save money and time, if we do different service actions at the same time.

5.2. Dynamic Programming

Dynamic Programming is a framework to solve problems where decisions are made in stages. The framework is closely connected to MDP from the last section. At each stage, the state of the system is observed, and an action influencing the system has to be made. The decided action deterministically or stochastically influences the state to

be observed at the next stage. Further, an immediate reward is gained. The goal is to maximize the reward and obtain a policy which gives us the best decision for every stage and state of our problem. The main challenge is to balance the decision policy in such a way that we get high actual and future reward. We strictly follow [10], [11] and [73].

Our basic (discrete) Dynamic Programming model consists of two parts:

1. A discrete time dynamic system which is given by

$$x_{t+1} = g_t\left(x_t, a_t, \omega_t\right),\ t = 0, 1, \ldots, T-1,$$

with

$t \in \mathbb{N}_0$	indexes the discrete time step,
$x_t \in \mathcal{S} \subseteq \mathbb{R}^{n_x}$	is the system state at time t from state space $\mathcal{S}_t$,
$a_t \in \mathcal{A}_t \subseteq \mathbb{R}^{n_a}$	is the control / decision variable to be selected at time step t from action space $\mathcal{A}_t$,
$\omega_t \in \mathbb{R}^{n_\omega}$	is a random parameter which represents uncertainty,
$T \in \mathbb{N}$	is the time horizon and
$g_t\left(x_t, a_t, \omega_t\right)$	is a function which characterize how the system state is updated.

2. A reward function $G_t\left(x_t, a_t, \omega_t\right)$ that is additive over time or in particular over every time step t. This means that the reward accumulates over time. As overall reward we get

$$G_T\left(x_T\right) + \sum_{t=1}^{T-1} G_t\left(x_t, a_t, \omega_t\right),$$

with some terminal reward $G_T\left(x_T\right)$ at the final time step T.

We assume for simplicity that our state space is discrete. In general we have uncertainty in our parameter ω_t in our system dynamic and reward function which underlies a proper probability distribution. In order to take uncertainty into account, we have to formulate our problem as optimization of the expected reward

$$\mathbb{E}\left[G_T\left(x_T\right) + \sum_{t=1}^{T-1} G_t\left(x_t, a_t, \omega_t\right)\right].$$

This gives us a risk neutral approach. We need two further inputs to complete our problem definition. The two missing inputs are:

1. Transition probabilities between states: We need a function which gives the probability that we change to state j from state i at time t, if we choose action a, i.e.

$$Pr\left(x_{t+1} = j | x_t = i, a_t = a\right) = Pr\left(j|i, a\right).$$

Since $x_{t+1} = g_t(x_t, a_t, \omega_t)$ depends on ω_t, our transition probability $Pr(j|i,a)$ is connected to the probability distribution of ω_t. Further, we remark that $Pr(j|i,a)$ depends only on the actual state. In particular, the transition probabilities are conditional probabilities.

2. We need the term of a policy or decision strategy which is a tuple of functions

$$\pi = (\pi_1, \ldots, \pi_T) \quad \text{with} \quad \pi_t \colon \mathcal{S} \mapsto \mathcal{A}_t.$$

The function π_t maps the state x_t to an action $\pi_t(x_t) = a_t$. The set $\mathcal{A}_t(x_t)$ consists of all allowable actions for the state x_t at time step t. A policy is called admissible, if $\pi_t(x_t) \in \mathcal{A}_t(x_t)$ holds for all states $x_t \in \mathcal{S}_t$ and for all time steps $t = 1, \ldots, T$. Further, we name with Π the set of all admissible policies.

We remark that in our definition the policy depends only on the actual state x_t at time t and not on history. We have to prove later that this policies are optimal, confer Theorem 5.3. With the term of a policy π we can rewrite our system dynamics as

$$x_{t+1} = g_t(x_t, \pi_t(x_t), \omega_t), \ t = 1, \ldots, T-1,$$

and the accumulated reward over the time horizon T as

$$V_\pi(x_0) = \mathbb{E}\left[G_T(x_T) + \sum_{t=1}^{T-1} G_t(x_t, \pi_t(x_t), \omega_t)\right].$$

We define an optimal policy $\pi^\star$ by

$$\pi^\star = \arg\max_{\pi \in \Pi} V_\pi(x_0)$$

We note that the optimal policy $\pi^\star$ depends on the initial state x_0. Further, the Dynamic Programming approach tries to find an optimal policy $\pi^\star$ which is optimal for every initial state x_0. We note that the optimal value $V_{\pi^\star}(x_0) = V^\star(x_0)$ depends on x_0 and is defined by

$$V^\star(x_0) = \max_{\pi \in \Pi} V_\pi(x_0).$$

We call $V^\star$ the optimal value function. Further, we interpret $V^\star$ as a mapping from initial states x_0 to optimal rewards $V^\star(x_0)$.

5.2.1. The Principle of Optimality

In this section we present the principle of optimality which is the main idea behind Dynamic Programming and it was discovered by Bellman in 1957 [8]. The principle of optimality means:

> "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision", confer [8].

Here, we take the theorem as formulated in [10]:

Theorem 5.3 (The Principle of Optimality)
Let $\pi^\star = (\pi_1^\star, \ldots, \pi_T^\star)$ be an optimal solution for the the Dynamic Programming problem, and assume that when using $\pi^\star$, a given state $x_{\hat{t}}$ occurs at time $\hat{t}$ with a positive probability. Consider the subproblem whereby we are at $x_{\hat{t}}$ at time $\hat{t}$ and wish to minimize the "reward to go" from time $\hat{t}$ to time T

$$\mathbb{E}\left[G_T\left(x_T\right) + \sum_{t=\hat{t}}^{T-1} G_t\left(x_t, \pi_t\left(x_t\right), \omega_t\right)\right].$$

Then the truncated policy $\left(\pi_{\hat{t}}^\star, \ldots, \pi_T^\star\right)$ is optimal for this subproblem.

Proof. The proof of the principle of optimality is very simple. If the truncated policy were not optimal as claimed, we would be able to switch the policy for our subproblem to one with higher reward in our initial problem, if we reach the state $x_{\hat{t}}$. This is a contradiction to our assumption that $\pi^\star$ is optimal. □

This principle is in our case the foundation to calculate the optimal maintenance policy. It shows that $\pi_t\left(x_t\right)$ does not depend on the history.

5.2.2. The Backward Algorithm

Next, we present an algorithm to solve the Dynamic Programming problem in an exact way. It is based on the principle of optimality from Theorem 5.3. The algorithm steps backward through the time to find the optimal value function V_t and policy π_t, $t = T, \ldots, 1$. The following theorem from [10] gives us the theoretical foundation:

Theorem 5.4 (Backward-Algorithm or DP-Algorithm)
For every initial state x_0, the optimal reward $V^\star\left(x_0\right)$ of the basic problem is equal to $V_0\left(x_0\right)$, given by the last step of the following algorithm, which proceeds backward in time from period $T-1$ to period 1:

$$V_T\left(x_T\right) = G_T\left(x_T\right)$$

$$V_t\left(x_t\right) = \max_{a_t \in \mathcal{A}_t(x_t)} \mathbb{E}\left[G_t\left(x_t, a_t, \omega_t\right) + V_{t+1}\left(g_t\left(x_t, a_t, \omega_t\right)\right)\right] \tag{5.5}$$

where the expectation is taken with respect to the probability distribution of ω_t, which depends on x_t and a_t. Furthermore, if $a_t^\star = \pi_t^\star\left(x_t\right)$ maximizes the right side of equation (5.5) for each x_t and t, then the policy $\pi^\star = (\pi_1^\star, \ldots, \pi_T^\star)$ is optimal.

The proof of Theorem 5.4 follows [10].

Proof. For any admissible policy $\pi = (\pi_1, \ldots, \pi_T)$ and each $t = 1, \ldots, T$, we denote $\pi^t = (\pi_t, \ldots, \pi_T)$. For $t = 1, \ldots, T$, let $V_t^\star(x_t)$ be the optimal reward for the $(T-t)$-stage-problem that starts at state x_t and time t, and ends at time T,

$$V_t^\star(x_t) = \max_{\pi_t} \mathbb{E}\left[G_T(x_T) + \sum_{t'=t}^{T-1} G_{t'}(x_{t'}, \pi_{t'}(x_{t'}), \omega_{t'})\right].$$

For $k = N$, we define $V_T^\star(x_T) = G_T(x_T)$. We will show by induction that the functions $V_t^\star$ are equal to the functions V_t generated by the DP algorithm, so that for $t = 0$, we will obtain the desired result.
Indeed, we have by definition $V_T^\star = V_T = G_T$. Assume that for one t and all x_{t+1}, we have $V_{t+1}^\star(x_{t+1}) = V_{t+1}(x_{t+1})$. Then, since $\pi^t = (\pi_t, \pi_{t+1})$, we have for all x_t:

$$\begin{aligned}
V_t^\star(x_t) &= \max_{\{\pi_t, \pi_{t+1}\}} \mathbb{E}\left[G_t(x_t, \pi_t(x_t), \omega_t) + G_T(x_T) + \sum_{t'=t+1}^{T-1} G_{t'}(x_{t'}, \pi_{t'}(x_{t'}), \omega_{t'})\right] \\
&= \max_{\pi_t} \mathbb{E}\left[G_t(x_t, \pi_t(x_t), \omega_t)\right. \\
&\left. + \max_{\pi_{t+1}} \left(\mathbb{E}\left[G_T(x_T) + \sum_{t'=t+1}^{T-1} G_{t'}(x_{t'}, \pi_{t'}(x_{t'}), \omega_{t'})\right]\right)\right] \qquad (5.6) \\
&= \max_{\pi_t} \mathbb{E}\left[G_t(x_t, \pi_t(x_t), \omega_t) + V_{t+1}^\star(g_{t+1}(x_t, \pi_t(x_t), \omega_t))\right] \qquad (5.7) \\
&= \max_{\pi_t} \mathbb{E}\left[G_t(x_t, \pi_t(x_t), \omega_t) + V_{t+1}(g_{t+1}(x_t, \pi_t(x_t), \omega_t))\right] \qquad (5.8) \\
&= \max_{a_t \in \mathcal{A}_t(x_t)} \mathbb{E}\left[G_t(x_t, a_t, \omega_t) + V_{t+1}(g_{t+1}(x_t, a_t, \omega_t))\right] \qquad (5.9) \\
&= V_t(x_t),
\end{aligned}$$

completing the induction. In equation (5.7), we used the definition of $V_{t+1}^\star$, and in equation (5.8) we used the induction hypothesis. In equation (5.9), we converted the maximization over π_t to a maximization over a_t, using the fact that for any function K of x and a, we have

$$\max_{\pi \in M} K(x, \pi(x)) = \max_{a \in \mathcal{A}_t(x)} K(x, a),$$

where M is the set of all functions $\pi(x)$ such that $\pi(x) \in A_t(x)$ for all x. □

The presented proof presents an interpretation for the function $V_t(x_t)$ as optimal value function for a $(T-t)$-stage problem starting at state x_t and time step t.

For a deeper insight in the DP-Algorithm, we reformulate it by taking the transition probability into account and remove the parameter ω_t from the reward function G_t. This case will fit better to our maintenance problem in the later section of the thesis. From now, we assume that the reward depends only on the actual state x_t and the

Algorithm 5.1 The Backward-Algorithm, confer [74].

function BACKWARD(T)
  for $x \in \mathcal{S}$ **do**
    $V_T(x) = \max_{a \in \mathcal{A}_T} G_T(x, a)$
  end for
  for t in $T-1$ to 1 **do**
    for $x \in \mathcal{S}$ **do**
      $V_t(x) = \max_{a_t \in \mathcal{A}_t(x_t)} \left(G_t(x_t, a_t) + \sum_{x \in \mathcal{S}} Pr(x|x_t, a_t) V_{t+1}(x) \right)$
    end for
  end for
  return $V_1, \ldots, V_T$
end function

action a_t an not on ω_t. We obtain

$$\begin{aligned} V_t(x_t) &= \max_{a_t \in \mathcal{A}_t(x_t)} \mathbb{E}\left[G_t(x_t, a_t) + V_{t+1}(g_{t+1}(x_t, a_t, \omega_t))\right] \\ &= \max_{a_t \in \mathcal{A}_t(x_t)} \left(G_t(x_t, a_t) + \mathbb{E}\left[V_{t+1}(g_{t+1}(x_t, a_t, \omega_t))\right]\right) \\ &= \max_{a_t \in \mathcal{A}_t(x_t)} \left(G_t(x_t, a_t) + \sum_{x \in \mathcal{S}} Pr(x|x_t, a_t) V_{t+1}(x) \right). \end{aligned} \tag{5.10}$$

Due to this assumption we need only to calculate the expected value of every value function $V_{t+1}(g_{t+1}(x_t, a_t, \omega_t))$ and we can write out equation (5.10). Further, we introduce at this point the Q-functions which are defined by

$$Q_{a_t,t}(x_t) = G_t(x_t, a_t) + \sum_{x \in \mathcal{S}} Pr(x|x_t, a_t) V_{t+1}(x). \tag{5.11}$$

A Q-function gives us the value of the actual and future expected reward $Q_{a_t,t}(x_t)$ for specific action a_t and state x_t. With the help of a Q-function we can rewrite our value function V_t from equation (5.10) as

$$V_t(x_t) = \max_{a_t \in \mathcal{A}_t(x_t)} Q_{a_t,t}(x_t).$$

With this formulation we are able to calculate our optimal value functions $V_t(x_t)$ in a recursive manner for every state x_t and time step t. To achieve this we have to create a lookup table which includes for every time step $t = 1, \ldots, T$ and every state $x_t \in \mathcal{S}$ the optimal value function $V_t(x_t)$. The lookup table has T columns and $|\mathcal{S}|$ rows. We summarize the Backward-Algorithm in Algorithm 5.1.

This lookup table approach leads to the main drawback in the DP-Algorithm: The curse of dimensionality. Assume that we want to calculate $V_{T-1}(x_t)$. In this case we have to calculate $V_t(x_t)$ for every x_t, because we need the value function $V_t(x_t)$ to calculate the expectation in $V_{t-1}(x_t)$. This can be a hard task, if the state space $\mathcal{S}$ is

very big. This is called the curse of dimensionality in the state space. There is also a curse of dimensionality in the time horizon, if T gets big as compared to the time step. Further, the action space $\mathcal{A}_t$ can suffer from the curse of dimensionality, if there are many actions to choose. The calculation of the expectation operator $\mathbb{E}(\cdot)$ in the value function V_t can be hard to compute, if the possible size of future states x_{t+1} is large.

5.3. The Advanced Replacement Model

The last Section 5.2 delivered the theoretical background for our modeling approach and now we formulate our advanced replacement model in terms of Dynamic Programming. In the next sections we present definitions of every piece of the DP framework to set up our maintenance scheduling model.

The Time Interval

As first step we divede the reviewed time interval $[0, T = T_{\text{end}}]$ of our maintenance scheduling problem into $T \in \mathbb{N}$ equal distributed stages, e.g. we get a set

$$\{t_0 = 0, t_1, \ldots, t_T = T_{\text{end}}\} \text{ with } t_{i+1} - t_i = \text{const for all } i = 1, \ldots, T-1.$$

To simplify notation we mention only the stages number $t = 1, \ldots, T$ instead of the real time points t_i.

The State Space

Next, we give the description of the state x_t of one gas turbine's component. In general we have

$$x_t \in \{-1\} \cup C =: \mathcal{S}, \quad C \subset \mathbb{R}_+.$$

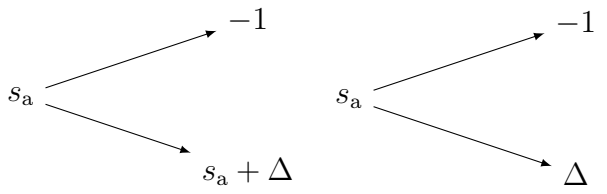
The set C is finite and it includes all possible ages of the component in terms of EOH, EBH, S or ES. It represents the consumed life of the component. The state $x_t = -1$ gives us the information that the component is failed. If our gas turbine consists of m components, then we define the complete state space $\mathcal{S}$ as

$$\mathcal{S} := \mathcal{S}_1 \times \ldots \times \mathcal{S}_m. \tag{5.12}$$

The Action Space

Our action space $\mathcal{A}$ for one component is very straight forward defined by

$$\mathcal{A} := \{0, 1\},$$



(a) Transition graph for a_0 (b) Transition graph for a_r

Figure 5.2.: The transition graphs for all actions $a \in \mathcal{A}$.

where $a = 1$ encodes to replace the component and $a = 0$ to do nothing. If a component is failed, we only can replace it. Therefore, $\mathcal{A}$ depends on the actual state x_t. The complete action space for the gas turbine is defined like in equation (5.12) by

$$\mathcal{A} := \mathcal{A}_1 \times \ldots \times \mathcal{A}_m.$$

The State Dynamics

The transition function $g_t(x_t, a_t, \omega_t) = x_{t+1}$ for the state of a component is defined by

$$g_t(x_t, a_t, \omega_t) := \begin{cases} x_{t+1} = x_t + \Delta c_t, & a_t = 0,\ \omega_t = 0 \\ x_{t+1} = \Delta c_t, & a_t = 1,\ \omega_t = 0 \\ x_{t+1} = -1, & a_t = \{0, 1\},\ \omega_t = 1 \end{cases},$$

where Δc_t is the amount of consumed life between t and $t+1$. In Figure 5.2 we illustrated the transition behavior. The state transition function for a gas turbine which consists of m components is given by

$$g_t(x_t, a_t, \omega_t) := (g_{t1}(x_{t1}, a_{t1}, \omega_{t1}), \ldots, g_{tm}(x_{tm}, a_{tm}, \omega_{tm})).$$

We assume that all component are independent of each other.

Transition Probabilities

Next, we show our model for the transition probability of a single component. We assume that a survival distribution $S(x_t)$ of a component is given. Then we define the transition probabilities $Pr(x_{t+1}|x_t, a_t)$ by

- No replacement:

$$Pr(x_{t+1}|x_t, a_t = 0) = \frac{S(x_{t+1})}{S(x_t)}, \quad 0 < x_t < x_{t+1}$$

and

$$Pr\left(x_{t+1} = -1 | x_t, a_t = 0\right) = 1 - \frac{S\left(x_t + \Delta c_t\right)}{S\left(x_t\right)}, \quad 0 < x_t.$$

- Replacement:

$$Pr\left(x_{t+1} = \Delta c_t | x_t, a_t = 1\right) = S\left(\Delta c_t\right)$$

and

$$Pr\left(x_{t+1} = -1 | x_t, a_t = 1\right) = 1 - S\left(\Delta c_t\right).$$

The probability to change from $x_t = -1$ to $x_{t+1} = -1$ is included in the replacement case. For the case of no replacement, we have to use conditional probabilities. The transition probability for the complete gas turbine state with m components is defined by

$$Pr\left(x_{t+1} | x_t, a_t\right) := \prod_{i=1}^{m} Pr\left(x_{t+1i} | x_{ti}, a_{ti}\right).$$

The Reward Function

As last missing piece, we have to define the reward function of our model. As in our replacement model from Section 4.4, we assume that the costomer has a certain revenue I_t per time interval. The gas turbine operator has to pay a penalty fee $C_{\mathrm{R}ti}$, if component i fails. Further, we split the maintenance cost in two parts. The equipment cost $C_{\mathrm{ME}ti} > 0$ and assembly cost $C_{\mathrm{MA}ti} > 0$ per component i. Also, there are dismantle dependencies between the parts. For simplicity, we suppose that if we disassemble component i, then we have to dismantle all components $j < i$. We define the complete reward function for the complete gas turbine state as

$$G_t\left(x, a\right) = I_t + \sum_{i=1}^{m} C_{\mathrm{R}ti} \min\left(x_i, 0\right) - \sum_{i=1}^{m} C_{\mathrm{ME}ti} a_i - \max_i \left(a_i \sum_{j}^{i} C_{\mathrm{MA}tj} \right). \tag{5.13}$$

In equation (5.13) the term $\min\left(x_{ti}, 0\right)$ says that the operator pays the penalty only in the case of a failure and $\max_i \left(a_{ti} \sum_j^i C_{\mathrm{MA}j} \right)$ ensures that all dismantle cost are charged only once for our assumed assembly dependency.

Remarks

The size of $\mathcal{S}_i$ and therefore the size of $\mathcal{S}$ is directly influenced by the discretization of our time interval $[0, T_{\mathrm{end}}]$ or more in detail the number T of stages. Between two stages

t and $t+1$ we have a certain increase Δc_{ti} of life consumption for every component i. Therefore, we have to take all Δc_{ti} into account to create all possible states x_t. If we look at one component i and $\Delta c_{ti} = \text{const}$, then we have $T+1$ possible states for the component i. We have

$$\mathcal{S}_i = \{-1\} \cup \{j\Delta c_{ti} | j = 1, \ldots, T\}.$$

We can decrease this number by introducing a maximal life $x_{\max}$ for every component. On this cases we get additional restrictions to our action space $\mathcal{A}$. If $x_t = x_{\max}$, then we can choose only the service action of replacement. Further, the definition of the state space as Cartesian product presents the reason why our approach suffers from the curse of dimensionality.

5.4. Numerical Analysis of the Backward-Algorithm

In this section, we present the numerical implementation of our advanced replacement problem and we show numerical results of our various example models.

5.4.1. Numerical Implementation

The complete implementation was realized in R like our replacement model in Section 4.5.1. We used a multi cpu implementation through the R package "snow", confer [89]. The general structure of our implementation is the same as in our replacement model, confer Figure 4.10. We have the same three groups of input parameters technical, financial and model. Then we have to preprocess the input data such that it fits our discretization. The main part is to create a labeling that distinguishes all possible states $x_t \in \mathcal{S}$ which is needed for our lookup table. The box "solve optimization problem" is different compared to Section 4.5.1, because use a brute force parallelized search to solve

$$V_t(x_t) = \max_{a_t \in \mathcal{A}_t(x_t)} \left(F_t(x_t, a_t) + \mathrm{e}^{-i_{\text{no}}} \sum_{x \in \mathcal{S}} Pr(x|x_t, a_t) V_{t+1}(x) \right) \tag{5.14}$$

for every t and x_t. More precisely, it calculates the a cell of the lookup table which is associated with state x_t and time step t. The main input parameters are the time step t the value function V_{t+1} of the next time step and the state x_t. The max-operation is done by simple testing all possible actions $a_t \in \mathcal{A}_t(x_t)$. In our case this method is suitable, because the size of the action space $\mathcal{A}$ is relative small. For the parallel implementation, we split the state space $\mathcal{S}$ into #cpu=number of cpu sets $\tilde{\mathcal{S}}_i$, $i = 1, \ldots, \#\text{cpu}$ and we let each cpu solve the optimization problem in equation (5.14) for one set $\tilde{S}_i$. Finally, after our distributed calculations have terminated and transfered their results to a control

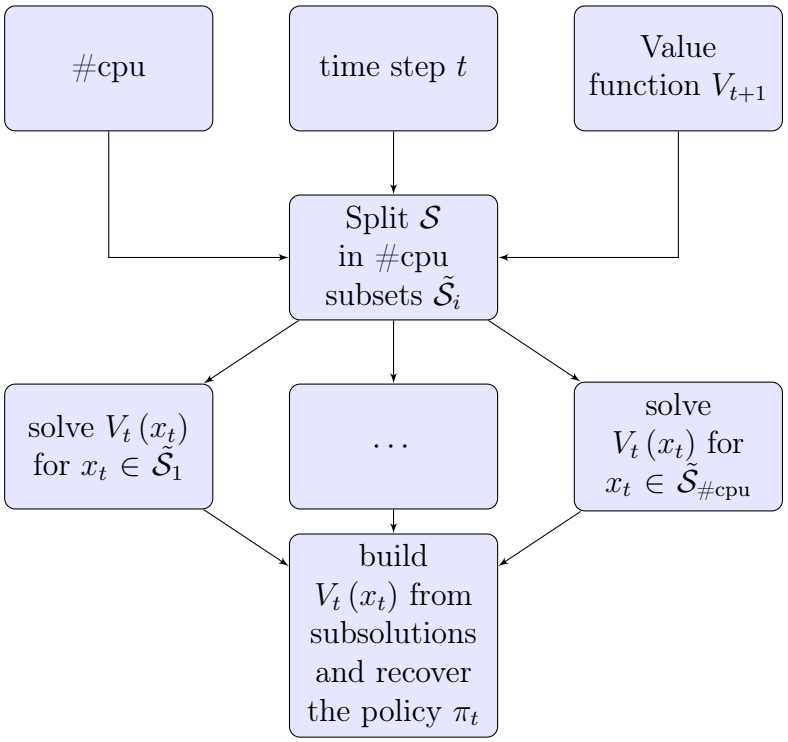


Figure 5.3.: Flow chart of the replacement model implementation.

node, we calculate

$$\arg \max_{a_t \in \mathcal{A}_t(x_t)} V_t(x_t)$$

to recover the optimal policy. Therefore, we have a lookup table for the optimal action $a_t \in \mathcal{A}$, too. This approach to parallelization only generates communication overhead when reporting the results to the the control node. Therefore, a very efficient scaling behavior can be expected in the number of cores.

5.4.2. Numerical Results

In this section we present numerical results for the various maintenance models for our Dynamic Programming approach. We will change the size of the state space in terms of number of components and in terms of possible state of one component, e.g. we increase $m = 1, \ldots, 5$ in $\mathcal{S} := \mathcal{S}_1 \times \ldots \times \mathcal{S}_m$ and $|\mathcal{S}_i|$. The size of $\mathcal{S}_i$ indicates how fine we have discretized the time steps. If we assume that it is possible to start an outage every day, we need a fine discretization of the state space, because many states will be possible. Further, we will analyze the dependencies between used cpu cores and the runtime. Therefore, we test our implementation on a Linux machine equipped with 16 cpu cores (two Intel Xeon cpu's with 2.4GHz and 8 cores) and 64GB RAM.

Time duration:	12 years
Power:	300MW
Operating hours per year:	8030
Starts per year:	30
Revenue:	$3\frac{\text{Euro}}{\text{MWh}}$

Table 5.1.: Model parameters for Variation.

#	$C_{\text{R}i}$ [mioE]	$C_{\text{E}i}$ [mioE]	$C_{\text{A}i}$ [mioE]	η_i [kEOH]	m_i	c_i $\left[\frac{\text{kEOH}}{\text{OH}}\right]$	$c_{\max}$ [kEOH]
1	15.0	5.0	0.250	200.0	2.1	1.0	100.8
2	20.0	2.0	0.100	3100.0	2.3	2.0	100.8
3	5.0	1.0	0.050	250.0	2.2	1.5	60.0
4	50.0	3.0	0.150	1000.0	2.8	1.0	90.0
5	30.0	2.5	0.125	533.0	2.4	3.0	80.0

Table 5.2.: Component data for model parameters for Variation.

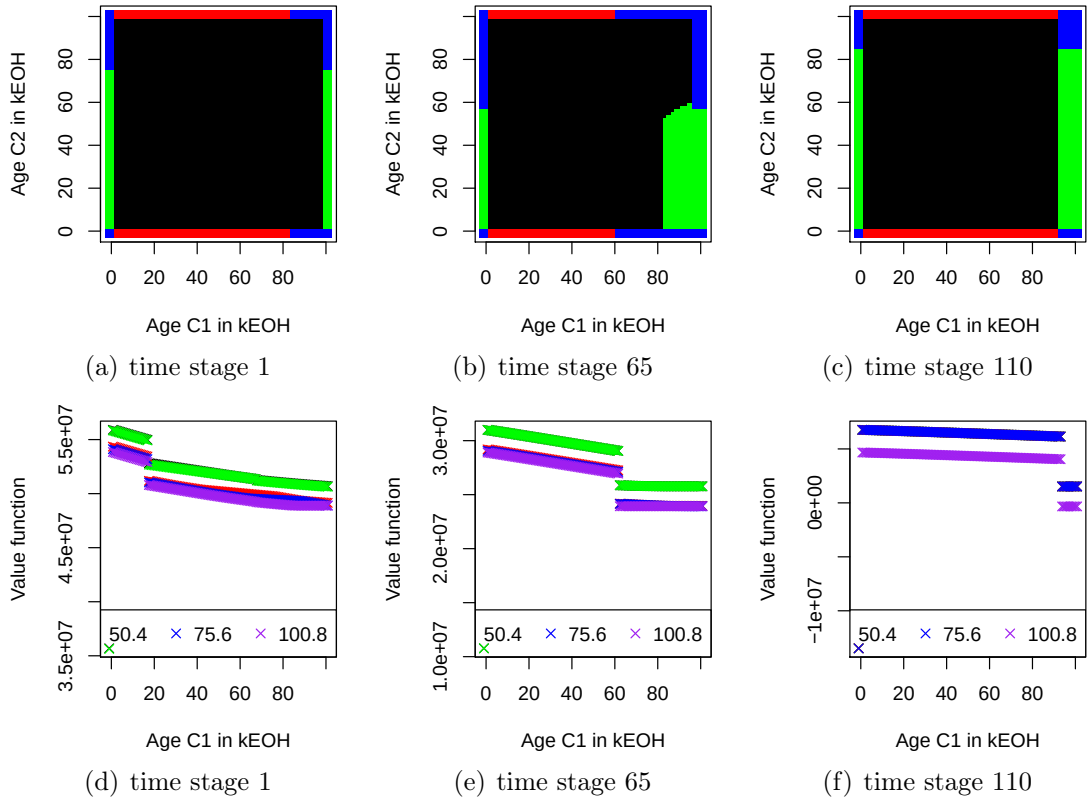
As last step, we compare the standard policy from section 4.5.2 against the optimal policy calculated by the Backward-Algorithm.

Variation of the Time Discretization and the Number of Counters/Components

In this paragraph, we analyze how the the computation time changes, if we increase the number of components and/or change time discretization. Further, we check how the number of cpu cores influences the runtime. The important model parameters are summarized in Table 5.1. We assume for every component a Weibull model for the failure distribution. Further, life consumption is constant over time, but every component accumulates life at a different speed $\frac{\partial}{\partial t}c_i$. The component specific data is presented in Table 5.2.

We start with a more detailed review of the two component model, because it will face up again in Sections 5.5 and 5.6. In Figure 5.4 we see the results of the two component model for a step size of 3 months. In Figures 5.4(a) to 5.4(c) we see the polices and in Figures 5.4(d) to 5.4(f) the associated value functions V_t for time points t_1, t_{65} and t_{110}. We recognize that the value function jumps at a certain state in all three cases and all not represented time points. This is reasonable, because there is always a critical state $x_{t\text{crit}}$ or age to replace a part and therefore we will replace it for all $x_t \geq x_{t\text{crit}}$. The jump point depends on the revenue I_t. We also see this behavior in the policies. There are structured areas where we do one distinct action.

Now, we start with our analysis of the runtime, if we change the number of components, the size of the discretization and use a different number of cpu cores. We discretize always the time span of 12 years in equal time steps Δt and therefore we have $T = \frac{12\text{year}}{\Delta t}$



(a) time stage 1 (b) time stage 65 (c) time stage 110

(d) time stage 1 (e) time stage 65 (f) time stage 110

Figure 5.4.: Results for the two component model. The first row shows the policy. Blue means two replace both components, green replace component 1, red replace component 2, black do nothing. The second row presents the value function for fixed states of component 2.

stages. In Table 5.3 we see the results. The runtime in brackets $[\cdot]$ represents the mean runtime per time step. We use this representation to eliminate the effect of the different number T of stages and to compare the runtime better. We do not calculate all possible parameter combinations, because for certain combinations the computation time is too long for example the case with 5 components and 1 cpu.

If we increase the number of cpu cores, then the runtime decreases. The scaling is not perfect, because if we increase the number of used cpu cores by the factor 2, then the runtime decreases only by the factor 1.5. Further, we see in Table 5.3 that the runtime increases linearly, if we keep the number of cpu cores and components constant and decrease the mesh size. In addition, the runtime increases if the size $|\mathcal{S}|$ of the state space keeps constant, but the number of components increases. The reason for this effect is that the state space becomes bigger. The size of the action space $\mathcal{A}$ doubles, if we add one component.

# components	T	Δt [months]	$\lvert\mathcal{S}\rvert$	runtime in seconds #cpu: 2	4	8	16
2	120	1	10585	3424.1 [28.5]	1981.8 [16.5]	1345.6 [11.2]	1019.1 [8.5]
	60	2	2701	444.7 [7.4]	257.8 [4.3]	197.4 [3.3]	157.2 [2.6]
	30	3	1225	104.1 [3.5]	58.2 [1.9]	47.4 [1.6]	44.9 [1.5]
	20	6	325	19.1 [1.0]	11.6 [0.6]	11.1 [0.6]	17.6 [0.9]
3	120	1	645685	– [–]	– [–]	– [–]	161955.0 [1349.6]
	60	2	83731	– [–]	– [–]	13350.0 [222.5]	10650.5 [177.5]
	30	3	25725	– [–]	– [–]	2149.7 [71.7]	1671.1 [55.7]
	20	6	3575	– [–]	– [–]	200.8 [10.0]	164.4 [8.2]
4	30	3	$1.183 \cdot 10^6$	– [–]	– [–]	– [–]	218228.3 [7274.3]
	20	6	$0.082 \cdot 10^6$	– [–]	– [–]	– [–]	10227.0 [511.4]
5	20	6	$0.657 \cdot 10^6$	– [–]	– [–]	– [–]	73408.6 [3670.4]

Table 5.3.: Runtime of the Backward-Algorithm for various m and Δt. The runtime in brackets $[\cdot]$ is the mean runtime per stage.

Comparison of the Optimal vs Standard Outage Policy

In this paragraph, we compare the standard maintenance policy from section 4.5.2 against the optimal one for different Δt. We replace component 1 at the age of 8000EOH, component 2 at the age of 25000EOH and component 3 at the age of 50000EOH. A Failure also leads to a replacement of a component. Further, we replace component 1 and 2, if we replace component 3. Also, we replace component 1, if we replace component 2. The results of our comparison are presented in Table 5.4. We compare the associated value functions $V_1^{\text{opt}}(x)$ and $V_1^{\text{sta}}(x)$ for a complete new gas turbine. On average the value function for the as news state connected to the optimal policy is 100.00Mio. Euro better than the standard policy as we see in Table 5.4.

ΔT in month	$V_1^{\text{opt}}(x)$ in Mio.Euro	$V_1^{\text{sta}}(x)$ in Mio.Euro
1	54.18	−91.79
2	54, 28	−66.70
3	43.65	−53.29
6	54.71	−68.29

Table 5.4.: Comparison of the Optimal vs Standard Outage policy in terms of the value function for the case that we start we a complete new gas turbine.

5.5. Approximate Dynamic Programing

The results in Table 5.3 show that the method does not scale very well, because it suffers from the curse of dimensionality. The state space grows fast, if we increase the number of components or decrease the step size Δt of the discretization. We extract this observation from Table 5.3. We need an approach to overcome this point. Therefore, we present the idea of approximate Dynamic Programing (ADP) in this section and we follow ideas from [73].

The LP-Algorithm

Before we start with the presentation of approximate Dynamic Programing, we introduce an algorithm which is based on linear programming to solve our Dynamic Programing problem instead of the Backward-Algorithm, confer [73]. We recapitulate the optimal value function

$$V_t(x_t) = \max_{a_t \in \mathcal{A}_t(x_t)} \left(F_t(x_t, a_t) + \mathrm{e}^{-i_{\text{no}}} \sum_{x \in \mathcal{S}} Pr(x|x_t, a_t) V_{t+1}(x) \right), \tag{5.15}$$

We obtain from equation (5.15) $|\mathcal{S}| \cdot |\mathcal{A}|$ linear inequalities in $V_t(x_t)$ for every time step which are given by

$$V_t(x_t) \geq F_t(x_t, a_t) + \mathrm{e}^{-i_{\text{no}}} \sum_{x \in \mathcal{S}} Pr(x|x_t, a_t) V_{t+1}(x) \text{ for all } x_t \in \mathcal{S},\ a_t \in \mathcal{A}_t(x_t). \tag{5.16}$$

We remark that the equality sign in equation (5.16) holds for the optimal action $a_t^\star$. Therefore, we interpret the inequalities in equation (5.16) as lower bounds of the value functions $V_t(x_t)$ in $x_t \in \mathcal{S}$. If we take the values of $V_t(x_t)$ as variables, we can a build a linear program with $|\mathcal{S}|$ variables and $|\mathcal{S}| \cdot |\mathcal{A}|$ linear inequalities. The complete linear program is given by

$$\min \sum_{x_t \in \mathcal{S}} V_t(x_t)$$

subject to

$$V_t(x_t) \geq F_t(x_t, a_t) + \mathrm{e}^{-i_{\text{no}}} \sum_{x \in \mathcal{S}} Pr(x|x_t, a_t) V_{t+1}(x) \text{ for all } a_t \in \mathcal{A}_t(x_t) \text{ and } x_t \in \mathcal{S}.$$

In the LP-Algorithm we also have to start at T and step backward trough the time t. The idea behind the algorithm is to find the biggest lower bound of equation (5.16) which is equal to the optimal value function $V_t^\star(x_t)$. One advantage is that we have only to solve T linear programs and not $T \cdot |\mathcal{S}|$ optimization problems to get the value functions.

We can recover the optimal action $a_t^\star$ for state x_t in the LP-Algorithm, if we search the inequality in equation (5.16) for the case where equality holds. The corresponding action a_t is the optimal one.

But the LP-Algorithm also suffers from the curse of dimensionality. Every linear program consists of $|\mathcal{S}|$ variables and there are $|\mathcal{S}| \cdot |\mathcal{A}_t|$ constraints. This can lead to very huge linear programs. So there is no big benefit to use the LP-Algorithm compared to the Backward-Algorithm, but it is a good starting point for the approximate Dynamic Programing Algorithm (ADP-Algorithm) presented in the following section.

5.5.1. ADP-Algorithm

As presented in [73], we show a method to solve Dynamic Programing problems in an approximative way. We start with the assumption that we can represent every value function $V_t(x_t)$ as

$$V_t(x_t) \approx \sum_{i=1}^{m} \beta_{ti}\phi_{ti}(x_t). \tag{5.17}$$

More precisely, we see $V_t(x_t)$ as a function of x_t and we assume that there exists a basis expansion of $V_t(x_t)$. The basis function can be chosen arbitrary for example splines or piecewise linear functions. The main requirement of the basis functions is that they represent the main features of $V_t(x_t)$ very well. Therefore, we have to choose the basis function problem specifically.

As next step we combine our LP-Algorithm with the basis evolution in equation (5.17). We obtain for the constraints in equation (5.16) new inequalities by

$$\sum_{i=1}^{m} \beta_{ti}\phi_{ti}(x_t) \geq F_t(x_t, a_t) + \mathrm{e}^{-i_{\mathrm{no}}} \sum_{x \in \mathcal{S}} Pr(x|x_t, a_t) \sum_{i=1}^{m} \beta_{t+1i}\phi_{t+1i}(x).$$

for all $x_t \in \mathcal{S}$ and $a_t \in \mathcal{A}_t(x_t)$. The objective function changes to

$$\min_{\beta_{ti}} \sum_{x_t \in \mathcal{S}} \sum_{i=1}^{m} \beta_{ti}\phi_{ti}(x_t) = \min_{\beta_{ti}} \sum_{i=1}^{m} \beta_{ti} \sum_{x_t \in \mathcal{S}} \phi_{ti}(x_t).$$

The variables changes from $V_t(x_t)$ to β_{ti}. Therefore, the number of variables in the linear program changes from $|\mathcal{S}|$ to the number of basis function m which reduces the problem size in terms of variables. But the number of constrains keeps unchanged. We can reduce the number of constraints problem specifically which we show in the next section.

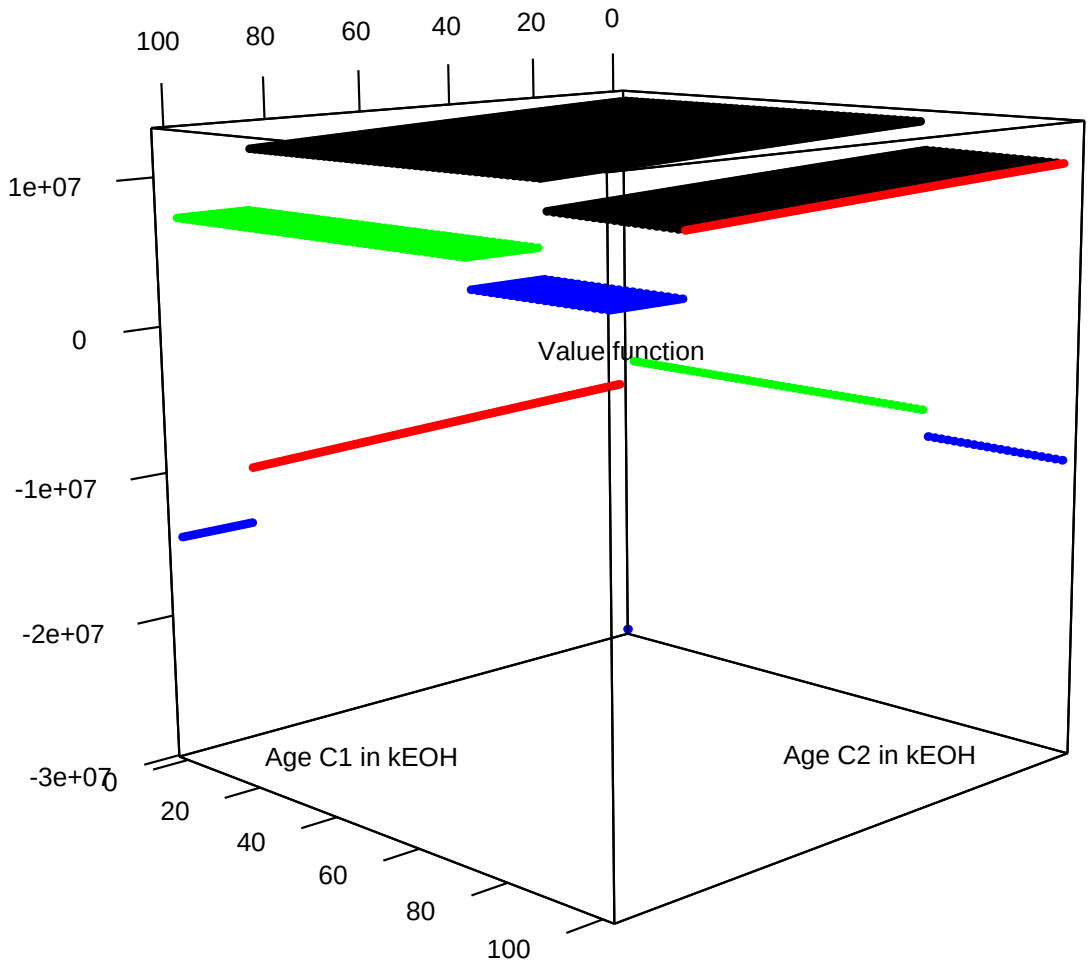


Figure 5.5.: 3d view of the value function $V_{100}(x_{100})$ partition for the 2 component model from Section 5.4.

5.5.2. An ADP-Algorithm for the Advanced Replacement Model

As a showcase for approximate Dynamic Programing, we present how to build an ADP-Algorithm for the two component model from Section 5.4. In Figures 5.5 and 5.6 we see the value function V_{100} and policy π_{100} for time step $t = 100$ for our two component model. The value function is divided into different areas which are separated by jumps in the value function. This corresponds to the policy and two special states families. The special states are the cases, where one or both components reach their maximal life or they fail. We see the mentioned structure in all other time steps t and for different model parameters like the revenue I. We use this structure to build a good approximation $\tilde{V}_t$ of the value function V_t.

Therefore, we divide our state space into maximally thirteen subspaces $\tilde{\mathcal{S}}_1$ and approx-

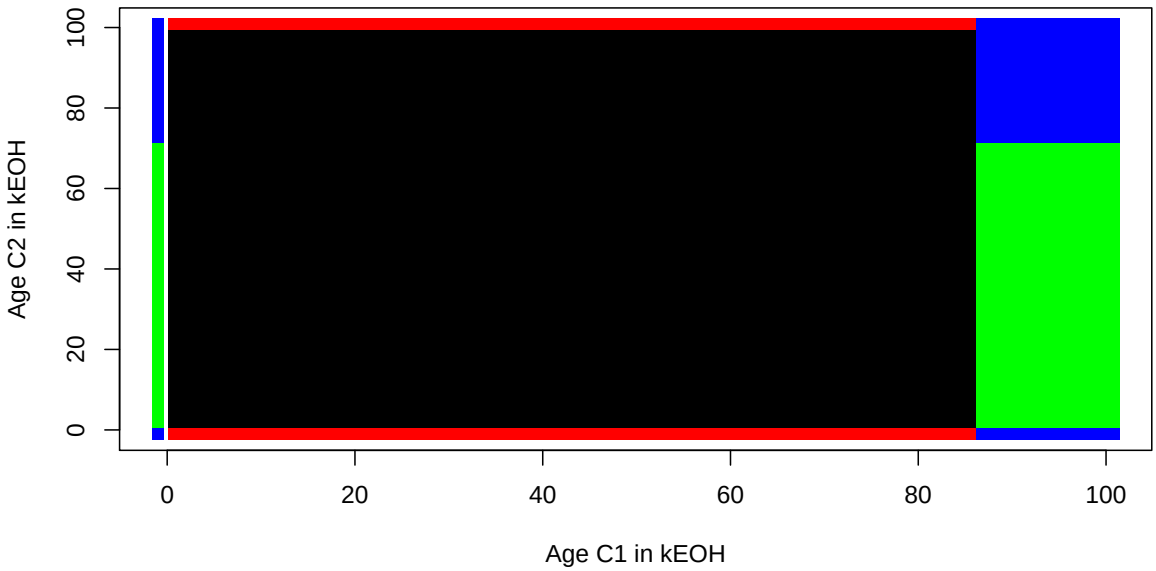


Figure 5.6.: Partition of the policy $pi_{100}k$ for the 2 component model from Section 5.4.

imate the value function on each of these subsets separately. The first four subsets $\tilde{\mathcal{S}}_i$, $i = 1, \ldots, 4$, are given by

$$\begin{aligned}
\tilde{\mathcal{S}}_1 &:= \{x \in \mathcal{S} | x_1 = -1\}, \\
\tilde{\mathcal{S}}_2 &:= \{x \in \mathcal{S} | x_2 = -1\}, \\
\tilde{\mathcal{S}}_3 &:= \{x \in \mathcal{S} | x_1 = \max(x_1)\}, \\
\tilde{\mathcal{S}}_4 &:= \{x \in \mathcal{S} | x_2 = \max(x_2)\}.
\end{aligned}$$

For these subsets we use our standard Backward-Algorithm to calculate the value function. This means we have to solve equation (5.14) $2|\mathcal{S}_1| + 2|\mathcal{S}_2|$ times. As the next step, we divide the remaining states

$$x \in \tilde{\mathcal{S}} := \mathcal{S} \setminus \left\{ \tilde{\mathcal{S}}_1 \cup \tilde{\mathcal{S}}_2 \cup \tilde{\mathcal{S}}_3 \cup \tilde{\mathcal{S}}_4 \right\}$$

into nine subsets $\tilde{\mathcal{S}}_i$, $i = 5, \ldots, 13$. Therefore, we search in the value function $V_t(x)$ restricted on the sets $\tilde{\mathcal{S}}_i$, $i = 1, \ldots, 4$ for jumps. In more detail we are looking for a change in the policy $\pi(x)$ restricted on $\tilde{\mathcal{S}}_i$, $i = 1, \ldots, 4$. To make this point clearer, we show it for example for the set $\tilde{\mathcal{S}}_2$. In this case we must always replace component 2 and we have to find the age of component 1 after which we replace component 1, too. In Figure 5.7 we see the corresponding value function V_t for time step $t = 100$. We replace component 1, if component 1 is older than approximately 83kEOH and component 2 is failed.

If we identified all four jump points $\tilde{x}_1, \ldots, \tilde{x}_4$ with $\tilde{x}_i = (\tilde{x}_{i1}, \tilde{x}_{i2})$, then we divide $\tilde{\mathcal{S}}$

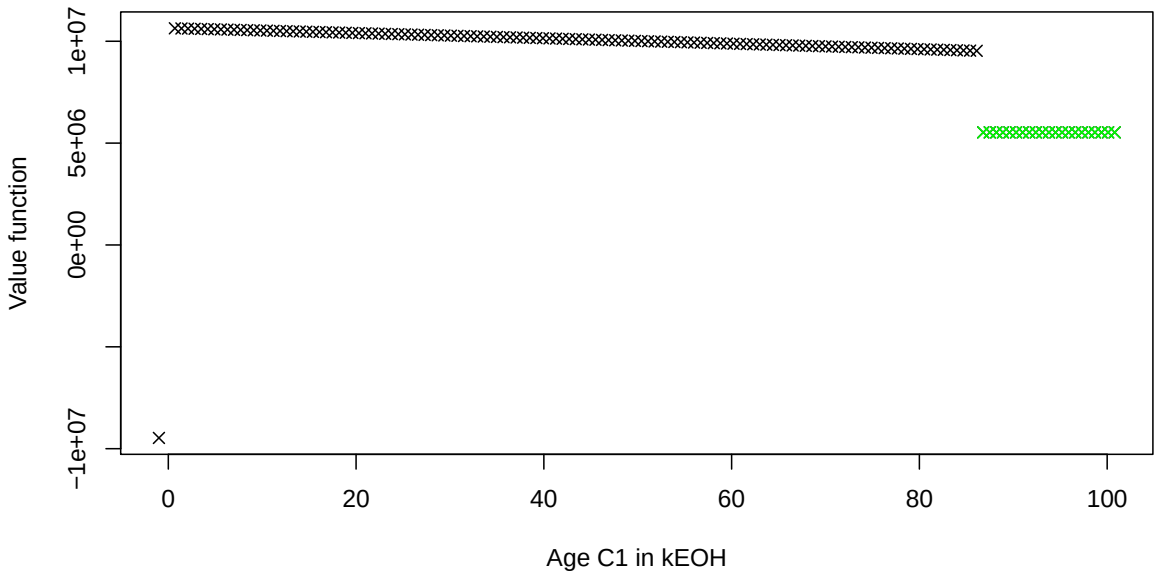


Figure 5.7.: Value function V_{100} restricted on $\tilde{\mathcal{S}}_2$. There is a clear jump at $x_{100,2} = 81.2$. The constant part where we replace the component is marked green.

into nine subsets. For example $\tilde{\mathcal{S}}_5$ and $\tilde{\mathcal{S}}_6$ are defined by

$$\begin{aligned}\tilde{\mathcal{S}}_5 &:= [\min(x_{t2}), \min(\tilde{x}_1, \tilde{x}_3)] \times [\min(x_{t1}), \min(\tilde{x}_2, \tilde{x}_4)] \quad \text{and} \\ \tilde{\mathcal{S}}_6 &:= [\min(\tilde{x}_1, \tilde{x}_3), \max(\tilde{x}_1, \tilde{x}_3)] \times [\min(x_{t1}), \min(\tilde{x}_2, \tilde{x}_4)] .\end{aligned}$$

In Figure 5.8 we illustrate the subdivisions schematically. Also, we see why we get nine subsets. This reason is why the jumps in $\tilde{\mathcal{S}}_1$ and $\tilde{\mathcal{S}}_3$ respectively $\tilde{\mathcal{S}}_2$ and $\tilde{\mathcal{S}}_4$ are not necessarily at the same point, e.g. in general $\tilde{x}_{12} \neq \tilde{x}_{32}$ and $\tilde{x}_{21} \neq \tilde{x}_{41}$.

Then, on the subsets $\tilde{\mathcal{S}}_5, \ldots, \tilde{\mathcal{S}}_{13}$ we use a local linear interpolation method to capture the function behavior in the different areas. Therefore, we create on every subset $\tilde{\mathcal{S}}_i$, $i = 9, \ldots, 13$ a mesh $\mathbb{G}_i$ by a latin hypercube sampling method from the R package "lhs", confer [17]. Further, we use the LP-Algorithm to determine the coefficients β_{tij} of the local linear interpolation approach which is implemented by the R package "geometry", confer [47] and [60] for more background information. As linear program solver we use "lp_solve", confer [9]. We use in the LP-Algorithm only $|\mathbb{G}_i|$ states $x_t \in \mathbb{G}_i \subseteq \tilde{\mathcal{S}}_i$ to create the linear inequalities in (5.16). Therefore, we define $|\mathbb{G}_i|$ local basis functions ϕ_{tj} with following property

$$\phi_{tj}(g_k) = \begin{cases} 1, & k = j \\ 0, & k \neq j \end{cases}, \quad g_k \in \mathbb{G}_i = \left\{g_1, \ldots, g_{|\mathbb{G}_i|}\right\}.$$

In summery we need only

$$2|\mathcal{S}_1| + 2|\mathcal{S}_2| + \sum_{i=5}^{13} |\mathbb{G}_i|$$

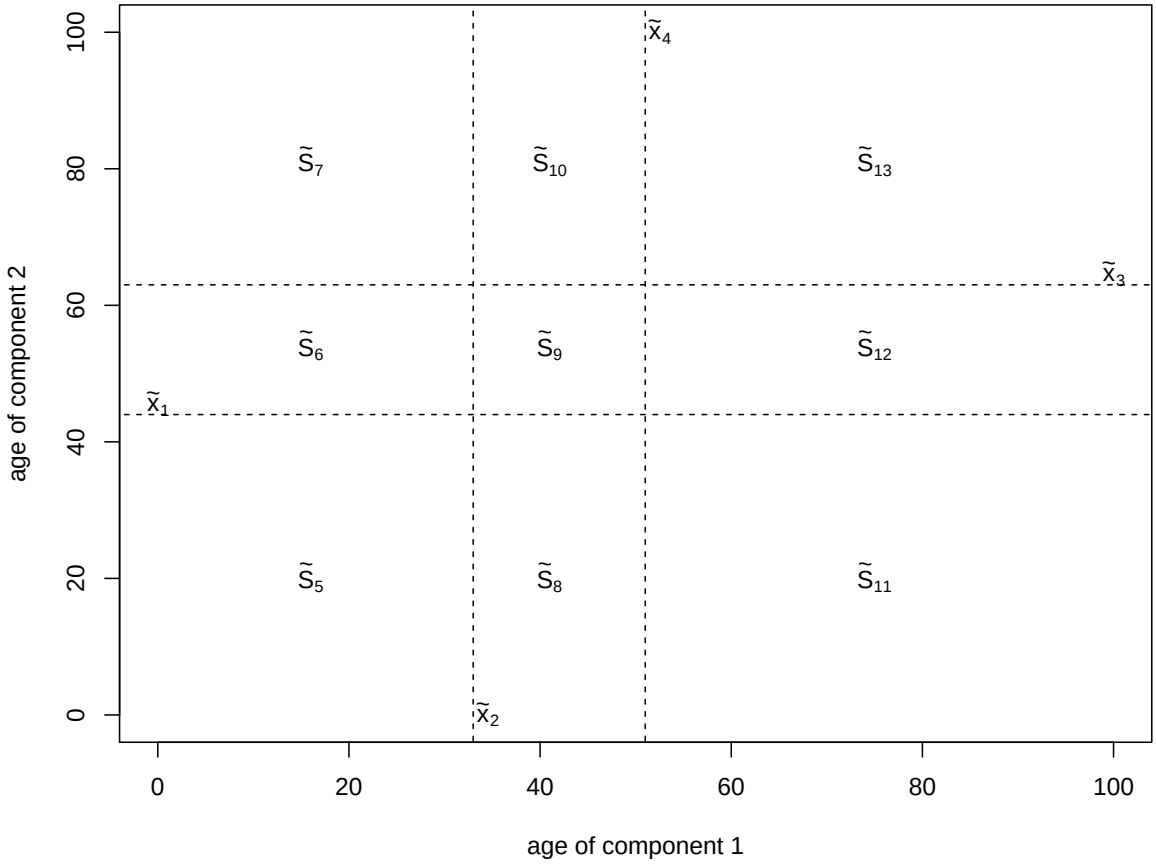


Figure 5.8.: Schematic subdividing of $\tilde{\mathcal{S}}$ according to the structure of V_t.

states to build a reasonable approximation $\tilde{V}_t$ of V_t. In the next section, we will show the associated numerical results.

5.6. Numerical Analysis of the ADP-Algorithm

Now, we present results how good we can approximate our advanced two component replacement model with our approximate Dynamic Programing method from Section 5.5.2. We will compare the value functions $\tilde{V}_t$ and the policy $\tilde{\pi}$ from the ADP-algorithm with the results V_t and π from the Backward-Algorithm. Therefore, we introduce two error measures val_{err} and pol_{err}. For the value function error val_{err} we calculate

$$val_{\text{err}}\left(\tilde{V}_t, V_t, x_t\right) := \left|1 - \frac{\tilde{V}_t\left(x_t\right)}{V_t\left(x_t\right)}\right| \in [0,1] \quad \text{for every } x_t \in \mathcal{S} \text{ and } t = 1, \ldots, T \tag{5.18}$$

and for the error in the policy pol_{err} we define

$$pol_{\text{err}}\left(\tilde{\pi}_t, \pi_t\right) := 1 - \frac{\sum_{i=1}^{|\mathcal{S}|} \hat{e}_i\left(\tilde{\pi}_t, \pi_t\right)}{|\mathcal{S}|} \in [0,1] \quad \text{for } t = 1, \ldots, T \tag{5.19}$$

where

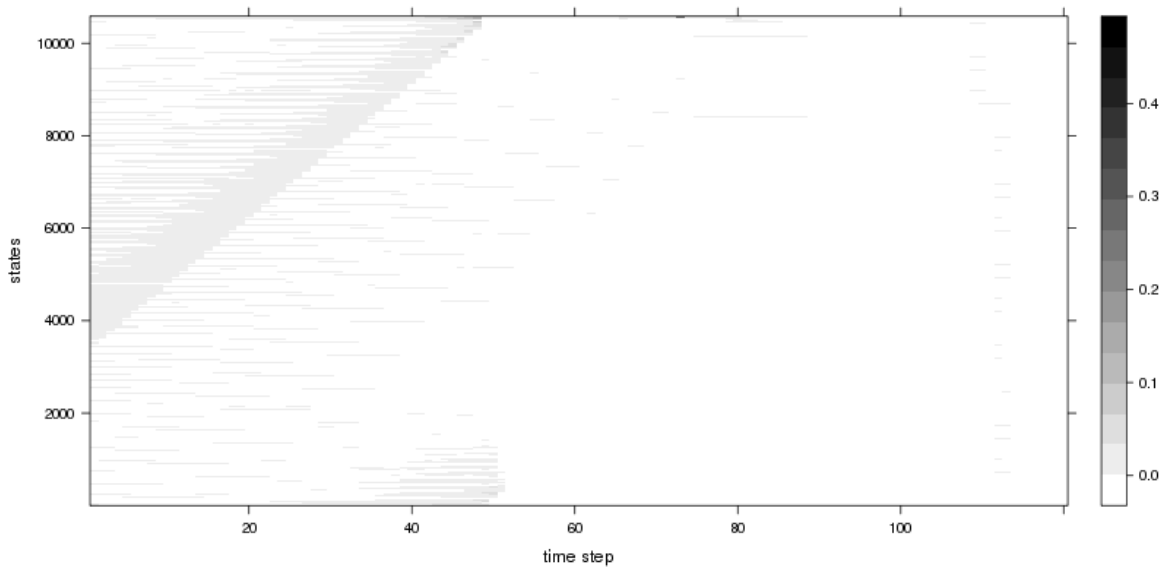
$$\hat{e}_i\left(\tilde{\pi}_t, \pi_t\right) := \begin{cases} 1, & \text{if } \tilde{\pi}_t\left(x_i\right) = \pi_t\left(x_i\right), \\ 0, & \text{else} \end{cases}$$

gives us the information that the policies deliver the same service action for the state $x_i \in \mathcal{S}$. The error in the approximated policy and value function is small, if pol_{err} respectively val_{err} is near zero.

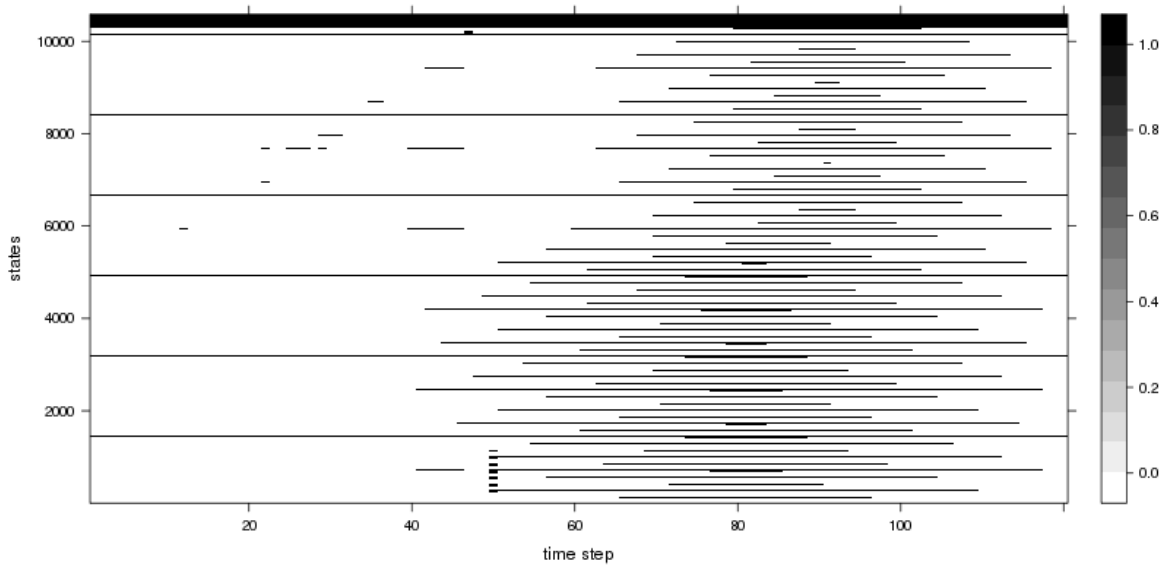
In Figures 5.9(a) and 5.9(b) we represent the error measures from equations (5.18) and (5.19) in a graphical way. The maximal error in the value function approximation $\tilde{V}_t$ is 0.46 in time step $t = 81$ for state $x_{81} = (38500, -1)$. The mean error is 0.0007. Further, we see that for 90.0% of the states $x_t \in \mathcal{S}$, $t = 1, \ldots, 120$, the error is smaller than 0.001 and for 98.0% of the states $x_t \in \mathcal{S}$, $t = 1, \ldots, 120$, the error is smaller than 0.01. Therefore, we can assume that the value function approximation $\tilde{V}_t$ fits very well.

As next step, we analyze the error in the approximated policy $\tilde{\pi}$. The overall error pol_{err} according to equation (5.19) is 0.11. This means that in 11.0% of the states the resulting action pair (a_1, a_2) is wrong. For a more detailed review, we split the error of $\tilde{\pi}$ in the two components $\tilde{\pi}_1$ and $\tilde{\pi}_2$ which deliver the action for component one or respectively for component two. The graphical representation can be found in Figures 5.9(c) and 5.9(d). The overall error for $\tilde{\pi}_1$ is 0.09 and for $\tilde{\pi}_2$ is 0.05 according to equation (5.19). We see that the error in $\tilde{\pi}_2$ is bigger than in $\tilde{\pi}_1$. Further, we see in Figure 5.9(c) that for certain states the policy $\tilde{\pi}_1$ is always wrong.

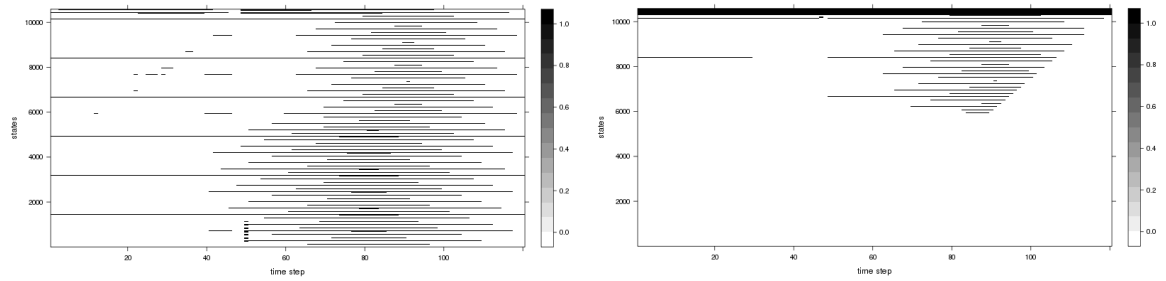
But in general, the quality of the approximation is very satisfying in view of the small runtime. The presented approximated solution $\tilde{V}_t$ was calculated in 275.5 seconds with the use of 2 cpu cores compared to 1019.1 respectively 3424.1 seconds of the Backward-Algorithm with the use of 16 respectively 2 cpu cores. But, we remark that we have to spend additional time to calculate the policy $\tilde{\pi}$ compared to the Backward-Algorithm which delivers policy lookup table simultaneously for all states $x_t \in \mathcal{S}$. But this is no drawback, because we can calculate the optimal action a_t for a state x_t in need. Also, the the calculation of the policy table in the Backward-Algorithm does not increase computational time significantly, because the policy table is an attachment of the optimization calculation in the value function lookup table. It saves only the result of the $\arg\max(\cdot)$ operator.



(a) Error in value function approximation $\tilde{V}_t$ according to equation (5.18).



(b) Error in policy approximation $\tilde{\pi}$ according to equation (5.19).



(c) Error in component 1 of the approximated policy $\tilde{\pi}$.

(d) Error in component 2 of the approximated policy $\tilde{\pi}$.

Figure 5.9.: Error in the approximated value function $\tilde{V}_t$ and policy $\tilde{\pi}$.

6. Inspection Model

There is one main drawback in our modeling approaches from Chapters 4 and 5. The models require that we know the actual state of the gas turbine's components exactly to schedule maintenance. This means, there is no uncertainty in the actual state and we can observe it perfectly. But in general, we cannot do this. For example, if we introduce a criterion like the presence of a crack to the state of a component, then we can see the crack only during an inspection, but not during normal operation. There are various components like turbine blades in the gas turbine which are only accessible during an outage. We need a special testing equipment like an ultrasonic testing device to find the crack.

Therefore, we need to extend our models with observability features and we apply the partially observable Markov Decision Processes (POMDP) for that reason. Further, the POMDP framework gives us the opportunity to establish the new service action type "inspection" which is often used in real world applications and connected to the case mentioned above.

During an inspection we gather information about the actual state of our component and we can leave the component as is or replace it prematurely to reduce the risk of a failure. In general the gained information during an inspection will change the probability of failure for the inspected component, because the gathered information lets us readjust the probability of a failure. We must use conditional probabilities which take the inspection criterion into account. Further, we have to establish a probability that a component fulfills one or more inspection criteria or not. With these probabilities we are able to model the state dynamics in a reasonable way. Therefore, we derive and present a simple inspection model. This model is based on crack growth.

Sirjaev and Dynkin started with research about POMDP, confer [33] and [84]. But as in the Dynamic Programming case the work of Wald from 1947 was important, confer [93]. Drake was the first person who developed the first explicit POMDP model in 1962, confer [30]. Astrom formulated independently at the same time finite horizon POMDP, confer [6]. Astrom also presented important results about the belief state of a POMDP. He proved that the belief state is a sufficient history. In 1971, Sondik presented essential results about the structure of the optimal solution and he established a solution algorithm based on this structure, confer [87]. This short overview is based on the work from [63]. In contrast to all presented examples in the overview, this work deals with a problem with a big state space which makes the problem's solution more complex.

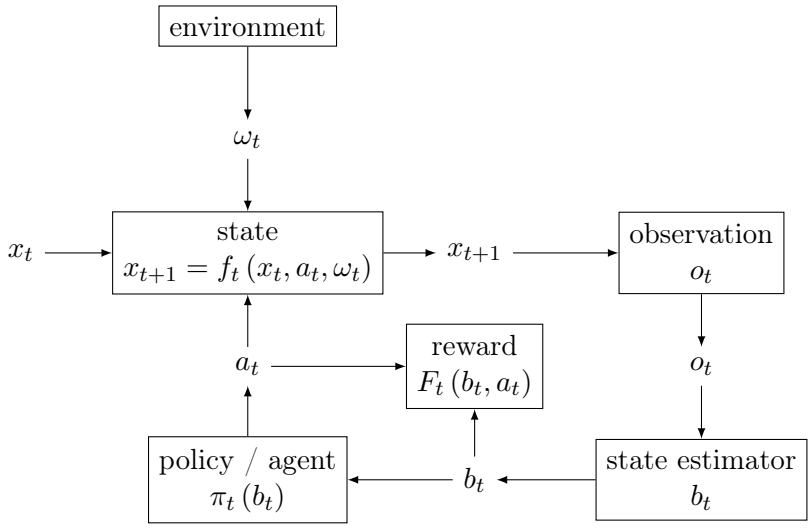


Figure 6.1.: Schematically representation of the POMDP Framework.

To establish the inspection model, we organize the chapter as follows: First, we extend the Markov Decision Processes from the last chapter to a partially observable Markov Decision Processes. Then, we present two solution approaches for partially observable Markov Decision Processes. One solution method, the IP-Algorithm, delivers the exact solution and the other one, α-min-Algorithm, an approximate solution. In the next step, we deduce a simple probabilistic inspection model for crack growth which includes inspection. Thereafter, we combine the POMDP approach with the crack growth model to set up our inspection model. As the last point, we present numerical results for our inspection model.

6.1. Partially Observable Markov Decision Process

Now, we establish a more general Markov Decision Process, a partially observable Markov Decision Process (POMDP), to overcome the problem that we do not always know the state of our system exactly. This will be true in our inspection model, where we do not know, if there exists a crack of a certain size. The basic properties of a POMDP and a MDP are quite the same. We have a state space, action space and state transition model based on the actual state and action. Further, the system behavior is like in a MDP, but we add a set of observations to the model. An observation becomes available to the agent after each state transition of the system, but it must not determine the complete state of the system. We illustrate the framework in Figure 6.1 and we define the POMDP through:

Definition 6.1 (POMDP)
A partially observable Markov Decision Process is defined as a tuple

$$\mathcal{M}_{PO} = \langle \mathcal{S}, \mathcal{A}, Pr, R, \mathcal{O}, O \rangle ,$$

where

- *the tuple* $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, Pr, R \rangle$ *is a Markov Decision Process,*
- $\mathcal{O}$ *is a finite set of observations the agent can see,*
- $O \colon \mathcal{S} \times \mathcal{A} \mapsto [0,1]$ *is the observation function, which gives for an action* $a \in \mathcal{A}$ *and a resulting state* $x \in \mathcal{S}$*, a probability to make observation* $o \in \mathcal{O}$*.*

In more detail, we define our observation function for a single observation o_t at time t by

$$Pr\left(o_t | x_t, a_{t-1}\right) := Pr\left(O_t = o_t | X_t = x_t, a_{t-1}\right)$$

where a_{t-1} is the action at time $t-1$ and x_t the resulting state at time t.

6.1.1. Policies and Belief States

To find an optimal policy of a POMDP we have to take the complete process history into account. In general it is not feasible to record the complete histories. Thus, we need other approaches to address this point. For MDP's we can use Markov policies which only use the actual state of the system. This type of policies contains the optimal policy as shown before, confer [6].

For POMDP we can not use the Markovian property and to take the complete history into account is more complex due to the observation space. We establish belief states to resolve this point. A belief state is a probability distribution over states. It is a summary statistic of the entire MDP history. This statistic is sufficient for the MDP history as shown in [6] and [86]. For that reason, an optimal policy can be calculated by the belief state instead of the complete history. It is true that the process over belief states is Markov, confer [6] and [86]. But we remark that our new state space is now continuous and it is defined by

$$\Pi\left(\mathcal{S}\right) := \left\{ b \in \mathbb{R}^{|\mathcal{S}|} | \sum_{i=1}^{|\mathcal{S}|} b_i = 1,\ b_i \geq 0 \right\}.$$

The i-th component of a belief state $b \in \Pi\left(\mathcal{S}\right)$ represents the probability that the system is in the i-th state.

Next, we derive a formula how the belief state b changes to b', if we are taking action a when making an observation o . We use Bayes formula, confer appendix A. We can

write $b'_{x'}$ as:

$$\begin{aligned} b'_{x'} =& Pr\left(x'|o,a,b\right) \\ =& \frac{Pr\left(o|x',a,b\right) Pr\left(x'|a,b\right)}{Pr\left(o|a,b\right)} \\ =& \frac{Pr\left(o|x',a\right) \sum_{x\in\mathcal{S}} Pr\left(x'|a,b,x\right) Pr\left(x|a,b\right)}{Pr\left(o|a,b\right)} \\ =& \frac{Pr\left(o|x',a\right) \sum_{x\in\mathcal{S}} Pr\left(x'|x,a\right) b_x}{Pr\left(o|a,b\right)}. \end{aligned} \tag{6.1}$$

Where the denominator $Pr\left(o|a,b\right)$ normalizes the resulting belief state to guarantee

$$\sum_{x'\in\mathcal{S}} b'_{x'} = 1$$

and is given by

$$\begin{aligned} Pr\left(o|a,b\right) &= \sum_{x'\in\mathcal{S}} Pr\left(o,x'|a,b\right) \\ &= \sum_{s'\in\mathcal{S}} Pr\left(x'|a,b\right) Pr\left(o|x',a,b\right) \\ &= \sum_{x'\in\mathcal{S}} \sum_{x\in\mathcal{S}} Pr\left(x|a,b\right) Pr\left(x'|x,a,b\right) Pr\left(o|x',a\right) \\ &= \sum_{x'\in\mathcal{S}} Pr\left(o|x',a\right) \sum_{x\in\mathcal{S}} Pr\left(x'|x,a\right) b_x. \end{aligned} \tag{6.2}$$

The calculations in equations (6.1) and (6.2) are taken from [54] and [58]. They show that we need only basic probability theory and the transition and observation probabilities to get the update formula for the new belief state. Further, we define $b^a_o \in \Pi\left(\mathcal{S}\right)$ as the belief state which results from taking action a and making observation o.

6.1.2. Value function

In this section we present how to calculate the optimal value function of a POMDP model. As in the MDP case, we can use the optimal value function to determine an optimal policy. In general, a non stationary t-step policy can be represented by a tree or a policy tree of depth t as presented in [54]. The top node of the policy determines the first action to be performed. Depending on the agent observation, the agent follows an arc of the tree to a node of the $t+1$ level which represents the next action.

Since the use of belief states gives us back the Markov property, we can use value function representation from the MDP / Dynamic Programming approach with a continuous states space $\Pi\left(\mathcal{S}\right)$ like shown in [19]. The action set $\mathcal{A}$ is the same as in the

original POMDP. The actual reward to be in belief state b and taking action a is

$$R_a^{\mathrm{T}} b = \sum_{x \in \mathcal{S}} R(x,a)\, b_x. \tag{6.3}$$

where $R(x,a)$ is the reward function from the underlying MDP. It is the expected reward over all states. Further, we define the set of possible successor belief states under action a by

$$\mathcal{B}(b,a) := \{b_o^a | o \in \mathcal{O}\}.$$

With this notation and the observation probability function $Pr(o|a,b)$, we obtain the following value function representation for our continuous belief state MDP by

$$V_{t-1}(b) = \max_{a \in \mathcal{A}_{t-1}} R_a^{\mathrm{T}} b + \mathrm{e}^{-i_{\mathrm{no}}} \sum_{o \in \mathcal{O}} Pr(o|a,b)\, V_t(b_o^a). \tag{6.4}$$

This formulation is equivalent to our classical value function formulation of a finite state MDP or Dynamic Programing method. We define for future use to equation (6.4) related value functions like in [18] as:

$$V_t^{a,o}(b) = \frac{1}{|\mathcal{O}|} R_a^{\mathrm{T}} b + \mathrm{e}^{-i_{\mathrm{no}}} Pr(o|a,b)\, V_{t-1}(b_o^a), \tag{6.5}$$

$$V_t^a(b) = \sum_{o \in \mathcal{O}} V_t^{a,o}(b), \tag{6.6}$$

$$V_t(b) = \max_{a \in \mathcal{A}_t} V_t^a(b). \tag{6.7}$$

The function $V_t^a(b)$ is the POMDP counterpart to the Q-functions in the classical MDP setting as presented in equation (5.11) from Section 5.2.2. $V_t(b)$ is equivalent to the value function in equation (6.4). The function $V_t^{a,o}(b)$ has quite the same interpretation as $V_t^a(b)$, but it includes only one observation $o \in \mathcal{O}$ and $\frac{1}{|\mathcal{O}|}$-th of the actual reward R_a.

The value functions of our continuous MDP has got a special structure namely it is a piecewise linear and convex (PWLC) function. The structure results from the converting of the POMDP to a MDP with a continuous state space. This observation does not hold in general for continuous MDP. Sondik proved the PWLC property in [87]. The property is very important, because it gives us the opportunity to represent the value function using a finite set $\mathcal{V}$ of vectors. The vectors in the set $\mathcal{V}$ are called α-vectors. The PWLC property is the starting point for many exact algorithms for finite horizon POMDP problems. To proof the PWLC property we need two basic properties of piecewise linear convex function:

Lemma 6.2 (PWLC Function)
Let f and g be two PWLC functions. Then $h_1 = af + bg$, $a, b \geq 0$, and $h_2 = \max(f,g)$ are PWLC functions.

Further, we conclude from Lemma 6.2 that the functions $V_t^{a,o}(b)$, $V_t^a(b)$ and $V_t(b)$ are PWLC functions. This result give us the possibility to prove Theorem 6.3 which was presented by Sondik in [87]:

Theorem 6.3 (PWLC Value Function)
The optimal value function $V_t(b)$ for a partial observable Markov Decision Process is a piecewise linear and convex function independent of the time step $t = 1, \ldots, T$.

Proof. We use induction over the remaining time horizon length T to proof Lemma 6.2. We start with the time horizon length $T = 1$. In this case only a single decision is remaining and we have not to take future rewards into account and we obtain

$$V_1^{a,o}(b) = \frac{1}{|\mathcal{O}|} R_a^{\mathrm{T}} b, \quad \text{for all } a \in \mathcal{A} \text{ and } o \in \mathcal{O}.$$

Thus, the value function $V_1^{a,o}(b)$ for every a and o is a linear function and therefore $V_1^{a,o}(b)$ is PWLC. The functions $V_1^a(b)$ and $V_1(b)$ are also PWLC as a result of Lemma 6.2.

For the induction step we assume that $V_{T-1}(b)$ is PWLC. Every PWLC function and especially $V_{T-1}(b)$ has the representation

$$V_{T-1}(b) = \max_{\alpha \in \mathcal{V}_{T-1}} \alpha^{\mathrm{T}} b,$$

where $\mathcal{V}_{T-1} \subset \mathbb{R}^{|\mathcal{S}|}$ is a finite set of vectors. We define

$$\tilde{\alpha}_{T-1}(b) = \arg \max_{\alpha \in \mathcal{V}_{T-1}} \alpha^{\mathrm{T}} b$$

and we get

$$V_{T-1}(b_o^a) = \tilde{\alpha}_{T-1}(b_o^a)^{\mathrm{T}} b_o^a.$$

Using this fact, the induction hypothesis and the representation of $V_T^{a,o}$ from equation (6.5), we obtain

$$\begin{aligned} V_T^{a,o}(b) &= \frac{1}{|\mathcal{O}|} R_a^{\mathrm{T}} b + \mathrm{e}^{-i_{\mathrm{no}}} Pr(o|a,b) V_{T-1}(b_o^a) \\ &= \frac{1}{|\mathcal{O}|} R_a^{\mathrm{T}} b + \mathrm{e}^{-i_{\mathrm{no}}} Pr(o|a,b) \tilde{\alpha}_{T-1}(b_o^a)^{\mathrm{T}} b_o^a \end{aligned}$$

and continuing calculations lead to

$$\begin{aligned} V_T^{a,o}(b) =& \frac{1}{|\mathcal{O}|} R_a^{\mathrm{T}} b + \mathrm{e}^{-i_{\mathrm{no}}} Pr(o|a,b) \tilde{\alpha}_{T-1}(b_o^a)^{\mathrm{T}} b_o^a \\ =& \frac{1}{|\mathcal{O}|} \sum_{x \in \mathcal{S}} R_{a,x} b_x + \mathrm{e}^{-i_{\mathrm{no}}} \sum_{x \in \mathcal{S}} \sum_{x' \in \mathcal{S}} b_x Pr(x'|x,a) Pr(o|x',a) \tilde{\alpha}_{T-1}(b_o^a)_{x'} \\ =& \sum_{x \in \mathcal{S}} b_x \left[\frac{1}{|\mathcal{O}|} R_{a,x} + \mathrm{e}^{-i_{\mathrm{no}}} \sum_{x' \in \mathcal{S}} Pr(x'|x,a) Pr(o|x',a) \tilde{\alpha}_{T-1}(b_o^a)_{x'} \right]. \end{aligned}$$

We define

$$\alpha_T^{a,o}(b,x) := \frac{1}{|\mathcal{O}|} R_{a,x} + \mathrm{e}^{-i_{\mathrm{no}}} \sum_{x' \in \mathcal{S}} Pr(x'|) Pr(o|x',a) \tilde{\alpha}_{t-1}(b_o^a)_{x'}$$

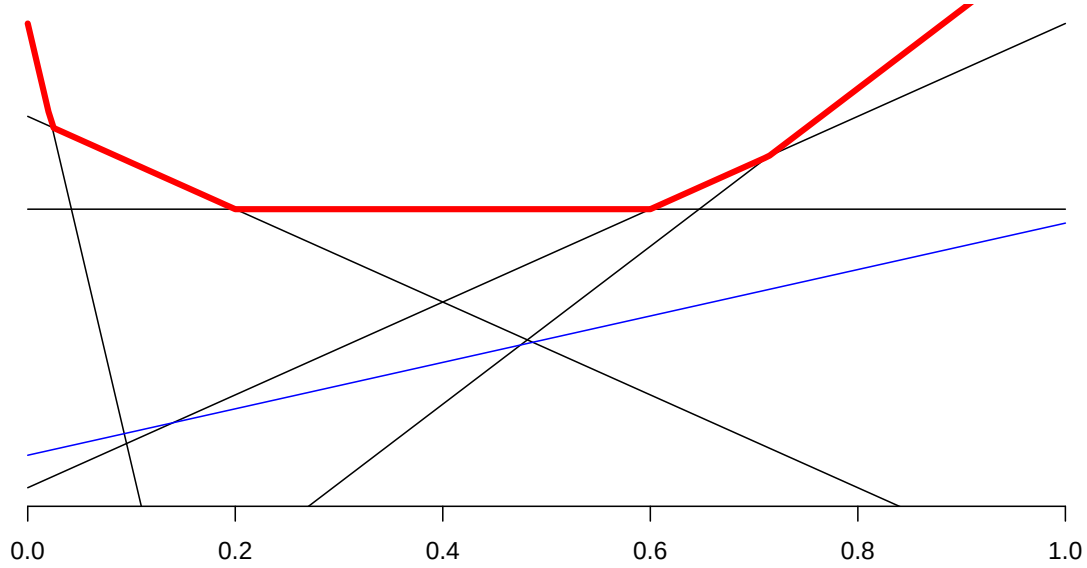


Figure 6.2.: The bold red line is the resulting PWLC function. For example the blue marked linear function is in no section dominant compared to the resulting one.

and it follows

$$V_T^{a,o}(b) = \alpha_T^{a,o}(b)^{\mathrm{T}} b.$$

Since the number of vectors $\alpha_T^{a,o}(b)$ is finite, we conclude again from Lemma 6.2 that $V_T(b)$ is PWLC. □

Every α-vector in the set $\mathcal{V}_t$ belongs to a policy tree and we can use it to determine the best action for the actual time step belief state pair.

For a two state example, we can represent the belief state (b_1, b_2) by one single number b_1, because we can make use of the constraint $b_1 + b_2 = 1$ and the associated value function $V(b) = \max_i g_i(b_1)$ is the max over a set over linear functions g_i in b_1. In Figure 6.2 we see such an example where the red line is the resulting PWLC function. We see that only three functions are active and every one belongs to one action. The blue function is never dominant. This observation implies that every PWLC function can be represented as

$$\max_{\alpha \in \mathcal{V}} \alpha^{\mathrm{T}} b \quad \text{where} \quad \mathcal{V} \text{ is a minimal set.}$$

In this case minimal set means that for every $\alpha \in \mathcal{V}$ there exists a belief state $b \in \Pi(\mathcal{S})$ such that $\alpha^{\mathrm{T}} b > \tilde{\alpha}^{\mathrm{T}} b$ for every $\tilde{\alpha} \in \mathcal{V} - \alpha$. The idea was proposed in [19].

Algorithm 6.1 DOMINATE checks if a vector $\alpha \in \mathcal{V}$ is dominant or not, confer [18].

```
function DOMINATE(α,V)
    L = (z, b) ← solve(LP(α, V))
    if IsInfeasible(L) or z ≤ 0 then
        return TRUE
    else
        return FALSE, b
    end if
end function
```

6.1.3. IP-Algorithm - The Exact Solution Algorithm for POMDP

The use of the standard value iteration algorithm is infeasible for a POMDP, because the set of belief states is infinite and not countable. Therefore, the PWLC property is the starting point for all exact POMDP algorithms, because the concept is always to calculate a minimal set of α-vectors. The simplest, but not efficient idea is to enumerate all possible α-vectors for one time step t in a set $\mathcal{V}_t$ and remove all $\alpha \in \mathcal{V}_t$ which are never active, confer [63]. The check for domination of an α-vector can be done through a linear program (LP $(\alpha, \mathcal{V})$) which is given by

$$\begin{aligned} & \max \; z \\ \text{subject to} \quad & \\ & (\alpha - \tilde{\alpha})^{\mathrm{T}} b > z \quad \text{for all } \tilde{\alpha} \in \mathcal{V} \\ & b \in \Pi(\mathcal{S}) . \end{aligned} \tag{LP$(\alpha, \mathcal{V})$}$$

If the optimal value $z^\star$ is greater zero, then the vector α is dominant in some belief state region and should not be pruned. If $z^\star$ is negative, then we have to prune the vector α from the set $\mathcal{V}$, e.g.

$$\mathcal{V} = \mathcal{V} \setminus \{\alpha\} .$$

The complete domination check method is presented in Algorithm 6.1.

Now, we present the Incremental-Pruning-Algorithm (IP-Algorithm) which was proposed in [18] and [98]. Basically the algorithm enumerates all possible α-vectors, but it prunes vectors during the generation to keep the possible size of α-vectors small. For a more detailed review we establish four notations. The cross sum of two vector sets A and B is defined as

$$A \oplus B := \{a + b \colon a \in A, b \in B\} .$$

Further, we define three vector sets which are associated to the α-vectors of our value

Algorithm 6.2 PRUNE reduces a set $\mathcal{V}$ to its minimal representation, confer [18].

function PRUNE($\mathcal{V}$)
  $S \leftarrow \emptyset$
  for $x \in \mathcal{S}$ **do**
    $\alpha \leftarrow \arg\max_{\bar{\alpha} \in \mathcal{V}} \bar{\alpha}^{\mathrm{T}} e_s$ ▷ e_i is the i-th unit vector
    $S \leftarrow S \cup \{\alpha\}$
    $\mathcal{V} \leftarrow \mathcal{V} \setminus \{\alpha\}$
  end for
  while $\mathcal{V} \neq \emptyset$ **do**
    $\alpha \in \mathcal{V}$
    $(sol, b) \leftarrow \text{DOMINATE}(\alpha, \mathcal{V})$
    if *sol* **then**
      $\mathcal{V} \leftarrow \mathcal{V} \setminus \{\alpha\}$
    else
      $\alpha \leftarrow \arg\max_{\bar{\alpha} \in \mathcal{V}} \bar{\alpha}^{\mathrm{T}} b$
      $S \leftarrow S \cup \{\alpha\}$
      $\mathcal{V} \leftarrow \mathcal{V} \setminus \{\alpha\}$
    end if
  end while
  return S
end function

function representations in equation (6.5) to equation (6.7). We define

$$\mathcal{V}_{t-1}^{a,o} := \text{prune}\left(\left\{\frac{1}{|\mathcal{O}|} R_a + \mathrm{e}^{-i_{\text{no}}} Pr\left(o|a,b\right) \alpha_t | \alpha_t \in \mathcal{V}_t\right\}\right) \tag{6.8}$$

$$\mathcal{V}_{t-1}^{a} := \text{prune}\left(\bigoplus_{o \in \mathcal{O}} \mathcal{V}_{t-1}^{a,o}\right) \tag{6.9}$$

$$\mathcal{V}_{t-1} := \text{prune}\left(\bigcup_{a \in \mathcal{A}} \mathcal{V}_{t-1}^{a}\right). \tag{6.10}$$

Equation (6.9) is comparable to the definition of the Q-function from equations (5.11) and (6.6).

The prune $(\cdot)$ function reduces a set of vectors to its minimal representation according to the definition from above. We use the DOMINATE operation from Algorithm 6.1 for the prune operation. The method to calculate $\mathcal{V}_{t-1}^{a}$ from $\mathcal{V}_{t-1}^{a,o}$ is taken from [63]. The IP-Algorithm focuses on an efficient method for calculating the set $\mathcal{V}_{t-1}^{a}$. We note that

$$\text{prune}\left(A \oplus B \oplus C\right) = \text{prune}\left(\text{prune}\left(A \oplus B\right) \oplus C\right)$$

and rewrite $\mathcal{V}_{t-1}^{a}$ as

$$\mathcal{V}_{t-1}^{a} = \text{prune}\left(\ldots \text{prune}\left(\text{prune}\left(\mathcal{V}_{t-1}^{a,o_1} \oplus \mathcal{V}_{t-1}^{a,o_2}\right) \oplus \mathcal{V}_{t-1}^{a,o_3}\right) \ldots \oplus \mathcal{V}_{t-1}^{a,o_m}\right)$$

Algorithm 6.3 Incremental-Pruning (IP): calculate $\mathcal{V}_t$ for all t.

function IP$(\alpha_1, \ldots, \alpha_n)$
  $\mathcal{V}_T \leftarrow \{\alpha_1, \ldots, \alpha_n\}$
  $\mathcal{V}_T \leftarrow \text{prune}\,(\mathcal{V}_T)$
  for $t = T-1, \ldots, 1$ **do**
    for $a \in \mathcal{A}$ **do** ▷ This loop will be used for a parallel implementation.
      for $o \in \mathcal{O}$ **do**
        $\mathcal{V}_t^{a,o} \leftarrow \text{prune}\left(\left\{\frac{1}{|\mathcal{O}|}R_a + \mathrm{e}^{-i_{\text{no}}}\, Pr\,(o|a,b)\,\alpha_{t+1} | \alpha_{t+1} \in \mathcal{V}_{t+1}\right\}\right)$
      end for
      $\mathcal{V}_t^a \leftarrow \text{prune}\left(\bigoplus_{o\in\mathcal{O}} \mathcal{V}_t^{a,o}\right)$
    end for
    $\mathcal{V}_t \leftarrow \text{prune}\left(\bigcup_{a\in\mathcal{A}} \mathcal{V}_t^a\right)$
  end for
  return $\mathcal{V}_t$, $t = T, \ldots, 1$
end function

with $m = |\mathcal{O}|$. We summarized the approach in Algorithm 6.3. Further, we note

$$\tilde{\mathcal{V}} := |\,\text{prune}\left(\mathcal{V}_{t-1}^{a,o_i} \oplus \mathcal{V}_{t-1}^{a,o_j}\right)| \geq \max\left(|\,\text{prune}\left(\mathcal{V}_{t-1}^{a,o_i}\right)|, |\,\text{prune}\left(\mathcal{V}_{t-1}^{a,o_j}\right)|\right).$$

It follows that $\tilde{\mathcal{V}}$ is monotonically non decreasing. According to [18], we get for the complexity

$$\mathcal{O}\left(|\mathcal{V}_t^a| \sum_o |\mathcal{V}_t^{a,o}|\right).$$

6.1.4. α-min-Algorithm - The Approximate Algorithm for finite Horizon POMDP

Based on the work in [32] we now propose an approximate algorithm to solve a finite horizon problem. The motivation is this: The exact algorithm tries to compute a minimal set $\mathcal{V}_t$ of α-vectors to get a representation of PWLC value function as $V_t\,(b) = \max_{\alpha_t\in\mathcal{V}_t} \alpha_t^{\mathrm{T}} b$. This task can be very hard, because the size of $\mathcal{V}$ is bounded by $|\mathcal{A}||\mathcal{S}|^{|\mathcal{O}|}$. Approximate algorithms like point-based approaches try to update the value function by a finite subset of sampled belief states $\tilde{\Pi} \subset \Pi\,(\mathcal{S})$. The belief states $b_k \in \tilde{\Pi}$ are chosen such that the corresponding $\alpha_t^{b_k}$-vectors are a good approximation of $\mathcal{V}_t$. We define

$$\tilde{\mathcal{V}}_t := \left\{\alpha_t^{b_k} | b_k \in \tilde{\Pi}\right\}.$$

In this case corresponding α-vector means that

$$\alpha_t^{b_k} = \arg \max_{\alpha_t \in P\left(\tilde{\mathcal{V}}_{t+1}\right)} \alpha_t^{\mathrm{T}} b_k,$$

where

$$P\left(\tilde{\mathcal{V}}_{t+1}\right) := \bigcup_{a\in\mathcal{A}} \mathcal{V}_t^a = \bigcup_{a\in\mathcal{A}} \bigoplus_{o\in\mathcal{O}} \mathcal{V}_t^{a,o} = \bigcup_{a\in\mathcal{A}} \bigoplus_{o\in\mathcal{O}} \left\{ \frac{1}{|\mathcal{O}|} R_a + \mathrm{e}^{-i_{\mathrm{no}}} Pr\left(o|a,b\right) \alpha | \alpha \in \mathcal{V}_{t+1} \right\}$$

is the back projection of $\tilde{\mathcal{V}}_{t+1}$ to time step t. We define the back projection operator $\mathrm{BL}\left(\tilde{V}_{t+1}\right)(\cdot)$ for one belief state b by

$$\mathrm{BL}\left(\tilde{V}_{t+1}\right)(b) := \max_{\alpha\in P\left(\tilde{\mathcal{V}}_{t+1}\right)} \alpha^{\mathrm{T}} b = \tilde{V}_t\left(b\right).$$

Further, we have

$$\tilde{\mathcal{V}}_{t+1} \subseteq \mathcal{V}_{t+1} \quad \text{and} \quad \tilde{\mathcal{V}}_{t+1} \subseteq P\left(\tilde{\mathcal{V}}_{t+1}\right) \subseteq P\left(\mathcal{V}_{t+1}\right).$$

Therefore, $\tilde{\mathcal{V}}_t$ is not necessary a minimal set. In general we define:

Definition 6.4
We call $\bar{\alpha}$ the vector function such that for every $b_k \in \Pi\left(\mathcal{S}\right)$

$$\bar{\alpha}\left(b_k\right) = \arg \max_{\alpha_t\in P\left(\tilde{\mathcal{V}}_{t+1}\right)} \alpha_t^T b_k.$$

A point based algorithm delivers very easily a lower approximation of the true value function for every time step. At the final time step $t = T$ we are setting $\tilde{\mathcal{V}}_T = \mathcal{V}_T$ and we calculate for every $t = T-1, \ldots, 1$ the following sets

$$\tilde{\mathcal{V}}_t = \left\{ \bar{\alpha}\left(b_k\right) | b_k \in \Pi\left(\mathcal{S}\right) \right\}$$

For every t, we have

$$\tilde{V}_t\left(b\right) = \max_{\alpha\in\tilde{\mathcal{V}}_t} \alpha^{\mathrm{T}} b \leq \max_{\alpha\in\mathcal{V}_t} \alpha^{\mathrm{T}} b = V_t\left(b\right).$$

Thus, we can calculate the maximal error err_t between $\tilde{V}_t$ and V_t through

$$err_t = \max_{b\in\Pi(\mathcal{S})} \mathrm{BL}\left(\tilde{V}_{t+1}\right)(b) - \tilde{V}_t\left(b\right). \tag{6.11}$$

Thus, our maximal error on $\tilde{V}_t$ for any t is given by

$$\sum_{k=t}^{T} err_k.$$

We will prove this later in Lemma 6.5. This error measurement is very common in different approximation methods like presented in [48].

Lemma 6.5 (Error in $\tilde{V}_t$)
Let $t = T-1, \ldots, 1$ and $\tilde{V}_{t+1}$ be a lower bound approximation of V_{t+1} with

$$\tilde{V}_{t+1}(b) \leq V_{t+1}(b) \leq \tilde{V}_{t+1}(b) + \epsilon \quad \textit{for all } b \in \Pi(\mathcal{S}).$$

Then,

$$\tilde{V}_t(b) \leq V_t(b) \leq \tilde{V}_t(b) + \epsilon \quad \textit{for all } b \in \Pi(\mathcal{S}),$$

where $\tilde{V}_t = \mathrm{BL}\left(\tilde{V}_{t+1}\right)$ and $V_t = \mathrm{BL}(V_{t+1})$.

We strictly follow the proof from [31]:

Proof. Let $b_t \in \Pi(\mathcal{S})$, α_t with $\alpha_t^{\mathrm{T}} b_t = V_t(b_t)$ and $\tilde{\alpha}_t$ with $\tilde{\alpha}_t{}^{\mathrm{T}} b_t = \tilde{V}_t(b_t)$. Let $a_t \in \mathcal{A}$ such that

$$\alpha_t = R_{a_t} + \sum_{o_{t+1} \in \mathcal{O}} \left(\alpha_{t+1}^{\alpha_t, o_{t+1}}\right)^{\mathrm{T}} M_{a_t, o_{t+1}},$$

$o_{t+1} \in \mathcal{O}$ and $\alpha_{t+1}^{\alpha_t, o_{t+1}} \in \mathcal{V}_{t+1}$. Further, let $\tilde{a}_t \in \mathcal{A}$ such that

$$\tilde{\alpha}_t = R_{\tilde{a}_t} + \sum_{o_{t+1} \in \mathcal{O}} \left(\tilde{\alpha}_{t+1}^{\tilde{\alpha}_t, o_{t+1}}\right)^{\mathrm{T}} M_{\tilde{a}_t, o_{t+1}},$$

where $\tilde{\alpha}_{t+1}^{\tilde{\alpha}_t, o_{t+1}} \in \tilde{\mathcal{V}}_{t+1}$ and $o_{t+1} \in \mathcal{O}$. We define $M_{a_t, o_{t+1}}$ and $M_{\tilde{a}_t, o_{t+1}}$ as $|\mathcal{S}| \times |\mathcal{S}|$-matrices

$$\begin{aligned} M_{a_t, o_{t+1}}(x_{t+1}, x_t) &= Pr(o_{t+1}|x_{t+1}, a_t)\, Pr(x_{t+1}|x_t, a_t) \quad \text{and} \\ M_{\tilde{a}_t, o_{t+1}}(x_{t+1}, x_t) &= Pr(o_{t+1}|x_{t+1}, \tilde{a}_t)\, Pr(x_{t+1}|x_t, \tilde{a}_t). \end{aligned}$$

We suppose first the case that $a_t = \tilde{a}_t$. We obtain

$$\begin{aligned} V_t(b_t) - \tilde{V}_t(b_t) &= \alpha_t^{\mathrm{T}} b_t - \tilde{\alpha}_t^{\mathrm{T}} b_t \\ &= \left(\sum_{o_{t+1} \in \mathcal{O}} \left(\alpha_{t+1}^{\alpha_t, o_{t+1}}\right)^{\mathrm{T}} M_{a_t, o_{t+1}} - \sum_{o_{t+1} \in \mathcal{O}} \left(\tilde{\alpha}_{t+1}^{\tilde{\alpha}_t, o_{t+1}}\right)^{\mathrm{T}} M_{\tilde{a}_t, o_{t+1}} \right)^{\mathrm{T}} b_t \\ &\leq \left\| \sum_{o_{t+1} \in \mathcal{O}} \left(\alpha_{t+1}^{\alpha_t, o_{t+1}} - \tilde{\alpha}_{t+1}^{\tilde{\alpha}_t, o_{t+1}}\right)^{\mathrm{T}} M_{a_t, o_{t+1}} \right\|_\infty \\ &:= \left\| \sum_{o_{t+1} \in \mathcal{O}} \Delta_{o+1}^{\mathrm{T}} M_{a_t, o_{t+1}} \right\|_\infty \end{aligned}$$

with the help of the Hölder inequality and $\|b_t\|_\infty \leq 1$. For every $o_{t+1} \in \mathcal{O}$ and $b_t \in \Pi(\mathcal{S})$ we get from the assumptions of Lemma that $\Delta_{o+1}^{\mathrm{T}} b_t \leq \epsilon$. Given that every standard basis vector e_i satisfies $e_i \in \Pi(\mathcal{S})$, we obtain

$$\begin{aligned} \|\Delta_{o+1}\|_\infty \leq \epsilon \quad &\text{for every } o_{t+1} \in \mathcal{O} \text{ and} \\ \Delta_{o+1}^{\mathrm{T}} M_{a_t, o_{t+1}} \leq E^{\mathrm{T}} M_{a_t, o_{t+1}}&, \end{aligned}$$

where $E = (\epsilon, \ldots, \epsilon)^{\mathrm{T}}$. Thus

$$V_t(b_t) - \tilde{V}_t(b_t) \leq \left\| E^{\mathrm{T}} \sum_{o_{t+1} \in \mathcal{O}} M_{a_t, o_{t+1}} \right\|_{\infty}.$$

Further, we have that $\sum_{o_{t+1} \in \mathcal{O}} M_{a_t, o_{t+1}} = T_{a_t}$ is stochastic and we get

$$\left\| E^{\mathrm{T}} \sum_{o_{t+1} \in \mathcal{O}} M_{a_t, o_{t+1}} \right\|_{\infty} \leq \epsilon.$$

Finally, we obtain

$$V_t(b_t) - \tilde{V}_t(b_t) \leq \epsilon.$$

Now, we choose $\tilde{a}_t \in \mathcal{A}$ arbitrary. Let be

$$\tilde{V}_t^{a_t}(b_t) = \left(R_{a_t} + \sum_{o_{t+1} \in \mathcal{O}} \left(\tilde{\alpha}_{t+1}^{a_t, o_{t+1}} \right)^{\mathrm{T}} M_{a_t, o_{t+1}} \right)^{\mathrm{T}} b_t.$$

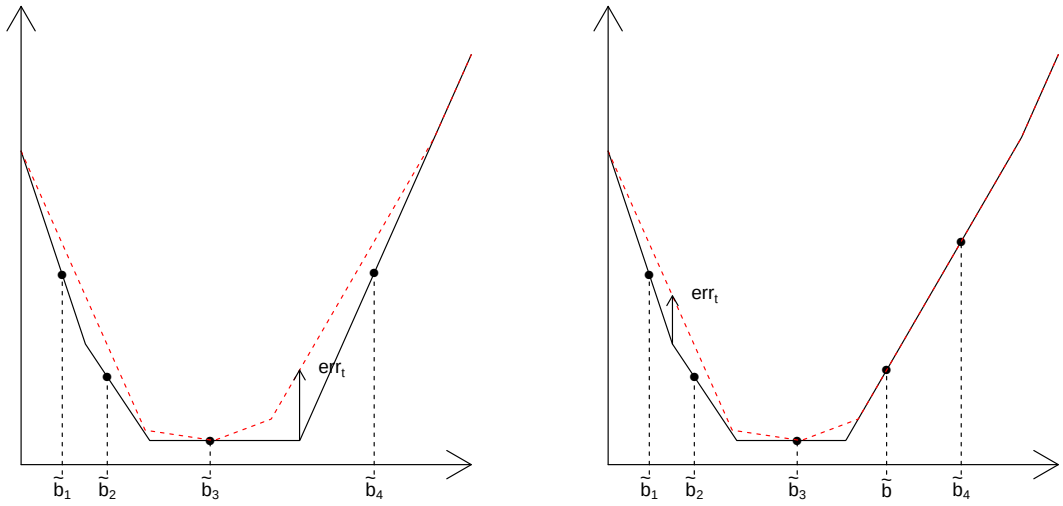
By definition of $\mathcal{V}_t(b_t)$, which is the maximum over $\tilde{a}_t$, we obtain that $\mathcal{V}_t(b_t) \geq \tilde{V}_t^{a_t}(b_t)$. Then, $\mathcal{V}_t(b_t) - \tilde{V}_t(b_t) \leq \mathcal{V}_t(b_t) - \tilde{V}_t^{a_t}(b_t) \leq \epsilon$. The demonstration of positivity of $\mathcal{V}_t(b_t) - \tilde{V}_t(b_t)$ is similar but easier and is therefore omitted. □

Later, we will use Lemma 6.5 to control the error in our Algorithm 6.6. The next equation

$$b^{\star} = \arg \max_{b \in \Pi(\mathcal{S})} \mathrm{BL}\left(\tilde{V}_{t+1}\right)(b) - \tilde{V}_t(b) \tag{6.12}$$

gives us important information to improve our approximation set $\tilde{\mathcal{V}}_t$. Equation (6.12) gives us the belief state $b^{\star}$ with the greatest error between the approximation and the exact solution. We can reduce the error by adding $\bar{\alpha}(b^{\star})$ to $\tilde{\mathcal{V}}_t$. If we solve equation (6.12), then we can add new vectors which are not dominated by other ones. This procedure is illustrated in Figure 6.3. But it is NP-hard to solve it directly. So, we are seeking for another way. As first step we reformulate equation (6.12) as a quadratic program (QP):

$$\begin{aligned} & \max\ g_t \\ & \text{subject to} \\ & \qquad g_t \leq \alpha_t^{\mathrm{T}} b_t - \tilde{\alpha}_t^{\mathrm{T}} b_t, \tilde{\alpha}_t \in \tilde{\mathcal{V}}_t \\ & \sum_{i=1}^{|\mathcal{S}|} b_{t,i} = 1 \\ & \qquad b_t \geq 0 \\ & \qquad \alpha_t \in P\left(\tilde{\mathcal{V}}_{t+1}\right). \end{aligned} \tag{QP}$$



(a) The error gap err_t marked by the arrow.

(b) Reduced error gap err_t after adding new α-vector $\bar{\alpha}\left(\tilde{b}\right)$ for belief point $\tilde{b}$.

Figure 6.3.: Illustration of the error gap in equation (6.12). The red dashed line refers to $V_t(b)$ and the black lines refers to $\tilde{V}_t(b)$ with $\tilde{\mathcal{V}}_t = \left\{\bar{\alpha}\left(\tilde{b}_1\right), \ldots, \bar{\alpha}\left(\tilde{b}_4\right)\right\}$.

b_t, α_t and g_t are the variables of (QP) and $\alpha_t^{\mathrm{T}} b_t$ is the α-vector formulation of the backward projection $\mathrm{BL}\left(\tilde{V}_{t+1}\right)(b_t)$ and the same holds for $\tilde{\alpha}_t^{\mathrm{T}} b_t$ with respect to $\tilde{\mathcal{V}}_t$. Also, this QP is not easy to solve, because it is not a concave QP and further the set $P\left(\tilde{\mathcal{V}}_{t+1}\right)$ is not given. Therefore, we reformulate the QP as a mixed integer linear program (MILP) which can be solved efficiently.

We start with a formulation of the set $P\left(\tilde{\mathcal{V}}_{t+1}\right)$. We need to find a set of linear inequalities to describe the convex hull C_t of $P\left(\tilde{\mathcal{V}}_{t+1}\right)$. We start with the following description of C_t:

$$C_t := \left\{\alpha_t \in \mathbb{R}^{|\mathcal{S}|} \colon \alpha_t^{\mathrm{T}} b_t \leq \bar{\alpha}_t \left(b_t\right)^{\mathrm{T}} b_t,\ b_t \in \Pi\left(\mathcal{S}\right),\ \alpha_t \geq 0\right\}.$$

This formulation uses an infinite number of inequalities to describe C_t. We can use a finite number of constraints to approximate C_t through a convex polyhedron

$$\tilde{C}_t^n := \left\{\alpha_t \in \mathbb{R}^{|\mathcal{S}|} \colon \alpha_t^{\mathrm{T}} b_t^i \leq \bar{\alpha}_t \left(b_t^i\right)^{\mathrm{T}} b_t^i,\ i = 1, \ldots, n,\ \alpha_t \geq 0\right\},$$

where $b_t^i \in \Pi\left(\mathcal{S}\right)$, $i = 1, \ldots, n$ with $n \geq |\mathcal{S}|$. It follows $C_t \subseteq \tilde{C}_t^n$ and we get a quadratic

program $\mathrm{QP}_n\left(b_t^1, \ldots, b_t^n\right)$:

$$\begin{aligned} & \max\ g_t^n \\ \text{subject to} & \\ & g_t^n \leq {\alpha_t^n}^{\mathrm{T}} b_t^n - \tilde{\alpha}_t^{nT} b_t^n, \tilde{\alpha}_t \in \tilde{\mathcal{V}}_t \\ & 1^{\mathrm{T}} b_t^n = 1 \\ & b_t^n \geq 0 \\ & \alpha_t^n \in \tilde{C}_t^n. \end{aligned} \tag{QP_n}$$

The solution of the new QP_n satisfies error bounds similar to Lemma 6.5. In particular we obtain from [31] and [32]:

Lemma 6.6
Let $\hat{\alpha}_t^n$, $\hat{\tilde{\alpha}}_t^n$, $\hat{b}_t^n$ and $\hat{g}_t^n$ be an optimal solution of QP_n. Then,

$$\max_{b_t \in \Pi(\mathcal{S})} \mathrm{BL}\left(\tilde{V}_{t+1}\right)(b_t) - \tilde{V}_t(b_t) \leq \hat{g}_t^n \leq \max_{b_t \in \Pi(\mathcal{S})} \mathrm{BL}\left(\tilde{V}_{t+1}\right) + \delta_t^n,$$

where $\delta_t^n = \hat{\alpha}_t^{nT} \hat{b}_t^n - \hat{\tilde{\alpha}}_t^{nT} \hat{b}_t^n$ at the optimum of QP_n.

Lemma 6.6 gives us the information, when should we expand our sampled belief state set $\tilde{\Pi}$ or not. We should expand the set, if δ_t^n is not small enough. Then we use $\hat{b}_t^n$ to construct a new constraint of $\tilde{C}_t^n$ and better approximation of C_t by $\tilde{C}_t^{n+1}$. For the convergence of the algorithm it is important that we can add new facets from the solution $\hat{\alpha}_t^n$ of QP_n which are not already in $\tilde{C}_t^n$. We call this procedure GenFacet $\left(\hat{\alpha}_t^n, \tilde{\mathcal{V}}_t + 1\right)$.

Next, we cite the important result from [31] and [32] that $\tilde{C}_t^n$ converges to C_t if we add facets through the solution of QP_n and GenFacet $\left(\hat{\alpha}_t^n, \tilde{\mathcal{V}}_t + 1\right)$.

Theorem 6.7 (Convergence of $\tilde{C}_t^n$)
$\tilde{C}_t^n$ converges to C_t. Thus, there exits $n^\star \in \mathbb{N}$ such that the solution $\hat{\alpha}_t^{n^\star}$ of QP_n belongs to C_t, i.e. $\delta_t^{n^\star} = 0$. For a given tolerance $\epsilon > 0$ there exits $n_\epsilon \in \mathbb{N}$ such that $\delta_t^{n_\epsilon} < \epsilon$.

Unfortunately, QP_n remains non concave and it is hard solve. But we know the structure of the optimal solution $\hat{\alpha}_t^n$. It belongs to a vertex of $\tilde{C}_t^n$ and it fulfills

$$\begin{aligned} & {\alpha_t^{\mathrm{T}}} b_t^i \leq \bar{\alpha}_t \left(b_t^i\right)^{\mathrm{T}} b_t^i,\ i = 1, \ldots, n, \\ & {\alpha_t^{\mathrm{T}}} b_t^{i_j} \leq \bar{\alpha}_t \left(b_t^{i_j}\right)^{\mathrm{T}} b_t^{i_j},\ j = 1, \ldots, |\mathcal{S}|. \end{aligned} \tag{6.13}$$

If we choose $b_t^i \in \tilde{\Pi}$ affinely independent, for example the corners of the simplex $\Pi(\mathcal{S})$, then we can represent $\hat{b}_t^n$ as convex combination of b_t^i and we get

$$\hat{b}_t^n \sum_{j=1}^{|\mathcal{S}|} \beta_j b_t^{i_j} \quad \text{with} \quad \sum_{j=1}^{|\mathcal{S}|} \beta_j = 1,\ \beta_j \geq 0.$$

The observation from the system (6.13) and the convex combination of $\hat{b}_t^n$ let us reformulate $\alpha_t^{\mathrm{T}} b_t$ as

$$\alpha_t^{\mathrm{T}} b_t = \sum_{j=1}^{|\mathcal{S}|} \beta_{i_j} \bar{\alpha} \left(b_t^{i_j} \right)^{\mathrm{T}} b_t^{i_j}. \tag{6.14}$$

Equation (6.14) gives us a linearization of the quadratic term $\alpha_t^{\mathrm{T}} b_t$ and we can reformulate QP_n as a mixed integer linear program (MILP_n):

$$\begin{aligned}
&\max\ g_t \\
&\text{subject to } g_t \leq W_t - U_t, \\
&\quad \sum_{j=1}^{|\mathcal{S}|} \beta_j = 1, \\
&\quad 1^{\mathrm{T}} b_t = 1, \\
&\quad b_t = \sum_{i=1}^{n} \beta_i b_t^i, \\
&\quad \alpha_t^{\mathrm{T}} b_t + y_i = \bar{\alpha}_t \left(b_t^i \right)^{\mathrm{T}} b_t^i, \quad i = 1, \ldots, n, \\
&\quad y_i \leq M \left(1 - \tilde{x}_i \right), \quad i = 1, \ldots, n, \\
&\quad U_t \geq \tilde{\alpha}_t^{\mathrm{T}} b_t, \quad \tilde{\alpha}_t \in \tilde{\mathcal{V}}_t, \\
&\quad \beta_i \leq \tilde{x}_i, \quad i = 1, \ldots, n, \\
&\quad \sum_{i=1}^{n} \tilde{x}_i \leq |\mathcal{S}|, \\
&\quad b_t \geq 0,\ y_i \geq 0,\ 0 \leq \beta_i \leq 1, \quad i = 1, \ldots, n, \\
&\quad \tilde{x}_i \in \{0, 1\}, \quad i = 1, \ldots, n, \\
&\quad W_t,\ U_t,\ g_t \geq 0,
\end{aligned} \tag{MILP_n}$$

where $M = \max_a (R_a)$. A further theorem from [31] gives us the equivalence of QP_n and MILP_n.

Theorem 6.8 ($\mathrm{QP}_n = \mathrm{MILP}_n$)
Let $b_t^i \in \Pi(\mathcal{S})$, $n \geq |\mathcal{S}|$. Problem $QP_n(b_t^1, \ldots, b_t^n)$ is equivalent to $MILP_n(b_t^1, \ldots, b_t^n)$.

This is the last missing result to create a procedure to solve equation (6.12) with a given error tolerance. We call the procedure FBB (Find Best Belief) which is represented in Algorithm 6.4. Further the number of operations to execute FBB is bounded by a finite number as shown in [31] and [32]. The result is included in the next lemma:

Lemma 6.9 (Complexity of FBB)
FBB *terminates in a finite number of iterations and it requires $O\left(P_N \times 2^{N+|\mathcal{S}|} + 1\right)$ operations in the worst case, where $N \in \mathbb{N}$ is the number of facets of C_t and P_N is a polynomial in N.*

Algorithm 6.4 Find the best belief state, confer [32]. Compared to [32] we added an iteration counter for the approximation of C_t.

function FBB(t, $\tilde{B}_t$, $\tilde{\mathcal{V}}_t$, $\tilde{\mathcal{V}}_{t+1}$, ϵ_p, $n_{\max}$)
  for n in $\{1, \ldots, |\mathcal{S}| + 1\}$ **do**
    $b_t^n = e_n$, ▷ $e_n \in \Pi(\mathcal{S})$ are unit vectors
  end for
  $n \leftarrow |\mathcal{S}| + 1$
  $n_{\text{iter}} \leftarrow 0$
  while $\Delta > \epsilon_p$ and $n_{\text{itr}} \leq n_{\max}$ **do**
    $\left(\alpha_t^n,\ \hat{b}_t,\ \hat{g}_t^n\right) \leftarrow \text{solve}\, MILP_n\left(b_t^1, \ldots, b_t^n\right)$
    $\Delta \leftarrow \delta_t^n = \hat{\alpha}_t^{nT} \hat{b}_t^n - \bar{\alpha}\left(\hat{b}_t^n\right)^{\mathrm{T}} \hat{b}_t^n$
    $F_t^n \leftarrow \text{GenFacet}\left(\hat{\alpha}_t^n,\ \tilde{\mathcal{V}}_{t+1}\right)$
    $b^{n+1} \leftarrow \hat{b}_t^n$
    $n \leftarrow n + 1$
    $n_{\text{iter}} \leftarrow \text{iter} + 1$
  end while
  return $\hat{g}_t^n$, $\hat{b}_t$, n_{itr}, Δ
end function

To have a control of the number of generated facets in GenFacet, we establish in Algorithm 6.4 an iteration counter n_{itr} compared to [32]. In this case it is possible that we do not reach the prescribed error tolerance ϵ_p.

A further result is that the Algorithm 6.5 or ϵ-min can be used to construct an approximative solution of our POMDP with maximal error ϵ equally distributed over all time steps, if we skip our introduced iteration limit n_{itr} for the facet generation. We obtain from [31] and [32]:

Theorem 6.10 (ϵ-min)
Let be $\epsilon > 0$. The ϵ-min-Algorithm solves approximately every finite horizon POMDP in finite time with maximum error ϵ, if we those $\epsilon_p \leq \epsilon$ in the ϵ-min-Algorithm.

The original ϵ-min-Algorithm is very powerful, because we solve any POMDP for a prescribed error tolerance which is uniformly distributed over all T time steps. But, the drawback is that there is no control over the number of α-vectors. In the worst case we will add a huge number of vectors to reach the prescribed error tolerance which can take a long time. Therefore, we present the α-min-Algorithm which gives control over the number of α-vectors at each time step and delivers a error bound. We present the detailed procedure in Algorithm 6.6. In summary we get the following result from [31] and [32]:

Theorem 6.11 (α-min-Algorithm)
The α-min-Algorithm solves approximately any POMDP with an arbitrary number N of

Algorithm 6.5 ϵ-min: Solve a given POMDP with maximum error tolerance ϵ equivalent distributed over all time steps $t = 1, \ldots, T_1$, confer [32].

function ϵ-min(T, $\mathcal{A}$, $\mathcal{S}$, $\mathcal{O}$, ϵ, ϵ_p, $n_{\max}$)
  $\tilde{V}_T \leftarrow \text{prune}\left(\{R_a | a \in \mathcal{A}\}\right)$
  $err \leftarrow= 0$
  for $t \in \{T, \ldots, 1\}$ **do**
    $\tilde{B}_t \leftarrow b_t^{\text{init}}$ ▷ Arbitrary $b_t^{\text{init}} \in \Pi(\mathcal{S})$
    $err_t \leftarrow \infty$
  end for
  for $t \in \{T-1, \ldots, 1\}$ **do**
    $n_{\text{itr}} \leftarrow 0$
    while $err_t > \frac{\epsilon}{T-1}$ and ($n_{\text{itr}} \leq n_{\max}$ and $\Delta \leq \epsilon_p$) **do**
      $(err_t,\ b_t^\star,\ n_{\text{itr}},\ \Delta) \leftarrow \text{FBB}\left(t,\ \tilde{B}_t,\ \tilde{\mathcal{V}}_t,\ \tilde{\mathcal{V}}_{t+1},\ \frac{\epsilon_p}{T-1},\ n_{\max}\right)$
      $\tilde{B}_t \leftarrow \tilde{B}_t \cup \{b_t^\star\}$
      $\tilde{\mathcal{V}}_t \leftarrow \tilde{\mathcal{V}}_t \cup \{\bar{\alpha}(b_t^\star)\}$
    end while
    $err = \sum_{t=1}^{T} err_t$
  end for
  return $\tilde{\mathcal{V}}_t$, $t = 1, \ldots, T$, err
end function

α-vectors per time step in finite time. Further, the α-min-*Algorithm provides an error bound between the optimal and approximate solution.*

But in general with this method we will not reach the prescribed error tolerance ϵ from the ϵ-min-Algorithm. The advantage of this algorithm over the ϵ-min-Algorithm is that we can balance the computation time and error bound with the two parameters ϵ and N.

6.2. Modeling Inspection

A further important service action is the inspection of a part. The modeling is more complex, than the modeling of replacement. We must take the information gain of the inspection into account. Basically, we gain the information whether the part fulfills the inspection criteria or not. For our further discussion we have only one inspection criterion which leads to a replacement if not met. First, we remark that we do not know the outcome of the inspection. There are two possible scenarios: We leave the part as is or we replace it by a new one. The modeling of the replacement is the same as before. We only reset the counter back to zero. If we leave the part as found, then we have to adapt the failure probabilities which need new considerations.

Algorithm 6.6 α-min: Solve POMDP with a maximum of N α-vectors and error tolerance ϵ_p, confer [32].

function α-min(T, $\mathcal{A}$, $\mathcal{S}$, $\mathcal{O}$, N, ϵ_p, $n_{\max}$)
  $\tilde{V}_T \leftarrow \text{prune}\left(\{R_a | a \in \mathcal{A}\}\right)$
  $err_T \leftarrow= 0$
  for $t \in \{T, \ldots, 1\}$ **do**
    $\tilde{B}_t \leftarrow b_t^{\text{init}}$ ▷ Arbitrary $b_t^{\text{init}} \in \Pi\left(\mathcal{S}\right)$
    $err_t \leftarrow \infty$
  end for
  for $t \in \{T-1, \ldots, 1\}$ **do**
    $n_{\text{itr}} \leftarrow 0$
    while $err_t > \frac{\epsilon}{T-1}$ and $|\tilde{\mathcal{V}}_t| < N$ and $\left(n_{\text{itr}} \leq n_{\max} \text{ and } \Delta \leq \frac{\epsilon_p}{T-1}\right)$ **do**
      $(err_t,\ b_t^\star,\ n_{\text{itr}},\ \Delta) \leftarrow \text{FBB}\left(t,\ \tilde{B}_t,\ \tilde{\mathcal{V}}_t,\ \tilde{\mathcal{V}}_{t+1},\ \frac{\epsilon_p}{T-1},\ n_{\max}\right)$
      $\tilde{B}_t \leftarrow \tilde{B}_t \cup \{b_t^\star\}$
      $\tilde{\mathcal{V}}_t \leftarrow \tilde{\mathcal{V}}_t \cup \{\bar{\alpha}\left(b_t^\star\right)\}$
    end while
    $err = \sum_{t=1}^{T} err_t$
  end for
  return $\tilde{\mathcal{V}}_t$, $t = 1, \ldots, T$, err
end function

At this point we introduce a model which describes the adaption of the failure probability due to an inspection.

6.2.1. A Simple Inspection Model Based on Mode I Crack Growth and Failure

We introduce a simple inspection model based on fracture mechanics. We start with a short repetition of basic principles of fracture mechanics and follow the ideas of [78].

In continuum based fracture mechanics a 3 dimensional (3d) crack can be modeled by a 2 dimensional separation of material. We can describe all possible 3d material openings by the linear combination of the three orthogonal crack opening modes I-III. The three modes are presented in Figure 6.4. We assume the existence of mode I cracks, because it gives us the largest crack growth for a cyclic stress-field in isotropic material. Further, we note that our approach describes only the observed phenomena and it can not describe the relevant mechanism. The relevant mechanism takes place at atomistic scale and we need only the effect and macro scale. Further, we assume that we describe the stress around the crack by linear elastic fracture mechanics and that our crack shape is a "penny" shape, e.g. it is circular.

We use the Paris' law to describe the crack growth under cyclic loading, confer [68]

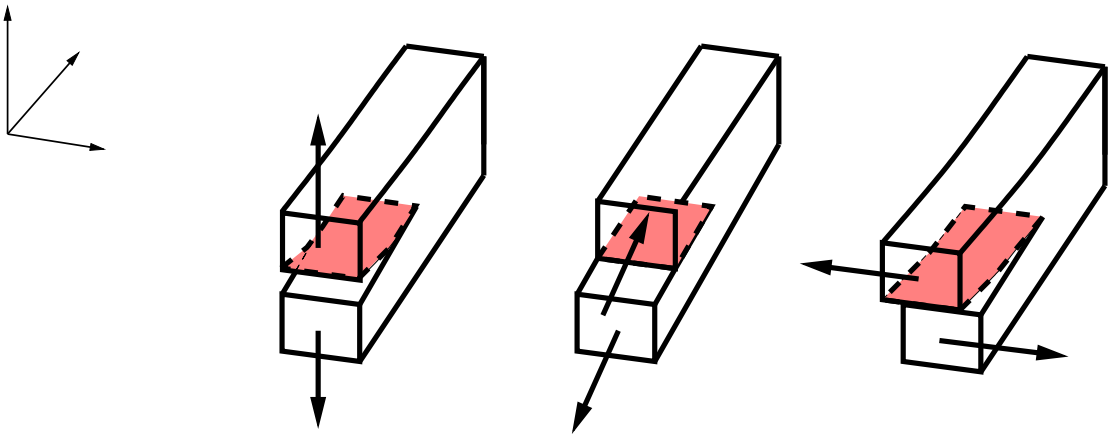


Figure 6.4.: Crack-modes I-III from the left to the right, cf. [78].

and [78]. It is given by

$$\frac{da}{dn} = AK_{\mathrm{I}}^{\bar{n}}. \tag{6.15}$$

It is a power law where a is the crack length, n is the cycle number, A and $\bar{n}$ are material constants and K_{I} is the stress intensity factor for mode I cracks which summarize the stress intensity around the crack depending on the crack shape and the load. For a penny shaped crack we get

$$K_{\mathrm{I}} = \sigma\sqrt{\pi a} \tag{6.16}$$

where σ is a tensile comparison stress. We sum up the two equations (6.15) and (6.16) and we obtain the following differential equation for mode I crack growth

$$\begin{aligned} \frac{da}{dn} &= A\pi^{\frac{\bar{n}}{2}}\sigma^{\bar{n}}a^{\frac{\bar{n}}{2}} \\ a\,(0) &= a_0. \end{aligned} \tag{6.17}$$

We get the solution of differential equation (6.17) by the method “separation of vari-

ables", [3]:

$$
\begin{aligned}
& \frac{da}{dn} = A\pi^{\frac{\bar{n}}{2}}\sigma^{\bar{n}}a^{\frac{\bar{n}}{2}} \\
\Rightarrow \quad & \frac{da}{a^{\frac{\bar{n}}{2}}} = A\pi^{\frac{\bar{n}}{2}}\sigma^{\bar{n}}dn \\
\Rightarrow \quad & \int_{a_0}^{a(n)} \frac{da}{a^{\frac{\bar{n}}{2}}} = \int_0^n A\pi^{\frac{\bar{n}}{2}}\sigma^{\bar{n}}dn + c \\
\Rightarrow \quad & \left[\frac{a^{1-\frac{\bar{n}}{2}}}{1-\frac{\bar{n}}{2}}\right]_{a_0}^{a(n)} = A\pi^{\frac{\bar{n}}{2}}\sigma^{\bar{n}}n + c \\
\Rightarrow & a\left(n\right)^{1-\frac{\bar{n}}{2}} - a_0^{1-\frac{\bar{n}}{2}} = \left(1-\frac{\bar{n}}{2}\right)A\pi^{\frac{\bar{n}}{2}}\sigma^{\bar{n}}n \ \ (c=0 \text{ initial condition}) \\
\Rightarrow \quad & a\left(n\right)^{1-\frac{\bar{n}}{2}} = \left(1-\frac{\bar{n}}{2}\right)A\pi^{\frac{\bar{n}}{2}}\sigma^{\bar{n}}n + a_0^{1-\frac{\bar{n}}{2}} \\
\Rightarrow \quad & a\left(n\right) = \left[\left(1-\frac{\bar{n}}{2}\right)A\pi^{\frac{\bar{n}}{2}}\sigma^{\bar{n}}n + a_0^{1-\frac{\bar{n}}{2}}\right]^{\frac{1}{1-\frac{\bar{n}}{2}}} =: \varphi_n\left(a_0,\sigma\right)
\end{aligned}
$$

where $\varphi_n\left(a_0,\sigma\right)$ is the flow map associated to Paris' law. The function $a\left(n\right)$ gives us the possibility to calculate the crack length a depending on the number of cycles n, the initial crack length a_0 and the stress field σ. If we know the critical crack length a_{crit}, then we are able to calculate the remaining number of cycles until failure in a deterministic manner. In general, we do not know the initial crack length a_0, because it is often unobservable. We try to describe this effect.

A Simplified Probabilistic Model for Failure

In contrast to Bolten, Gottschalk, Schmitz in [14] we assume that all cracks are in the worst possible position and failure is only possible in tensile stress mode I, see Figure 6.4. It should be a conservative assumption.

We need to define a tensile comparison stress σ_T for our calculations. Let σ be a stress tensor and σ_{I}, σ_{II} and σ_{III} the three principal stresses. We define

$$\sigma_{\text{T}} = \max\left(\sigma_{\text{I}}^+, \sigma_{\text{II}}^+, \sigma_{\text{III}}^+\right), \quad \sigma_J^+ = \max\left(\sigma_J, 0\right), \quad J = \text{I}, \text{II}, \text{III}$$

for all orientations $\vec{n} \in S^2 := \{x \in \mathbb{R}^2 \colon \|x\|_2 = 1\}$ of the crack surface.

As next step we need a failure criterion. Components fail, if the stress intensity factor K_{I} associated with a crack reaches a critical value. This value is the fracture toughness K_{I_c}. Then we obtain for the failure criterion:

$$K_{\text{I}_\text{c}} = \sigma_{\text{T}}\sqrt{\pi a_{\text{crit}}} \quad \Leftrightarrow \quad a_{\text{crit}} = \frac{1}{\pi}\left(\frac{K_{\text{I}_\text{c}}}{\sigma_{\text{T}}}\right)^2.$$

In general the shape Ω of a component with one or more cracks is a subset of the $\mathbb{R}^3$ like in Section 2.1.3. Therefore, the stress tensor σ will depend on the position $x \in \Omega$. We get $\sigma = \sigma(x)$ and as direct consequence our related comparison stress scalar field $\sigma_{\mathrm{T}}(x)$ depends on x. Further, let $\mathcal{C} = \Omega \times \mathbb{R}_+$ be the crack configuration space for a single crack

$$c = (x, a)$$

with the crack location $x \in \Omega$ and crack size $a \in \mathbb{R}_+$. For a given load situation $\sigma(x)$ the following set is the critical set of crack configurations

$$\mathcal{K}_{\mathrm{c}} = \left\{ (x, a) : a \geq \frac{1}{\pi} \left(\frac{K_{\mathrm{I_c}}}{\sigma_{\mathrm{T}}(x)} \right)^2 \right\}.$$

All cracks in $\mathcal{K}_{\mathrm{c}}$ will lead to a failure. Therefore, we need a possibility to estimate this number $|\mathcal{K}_{\mathrm{c}}|$. Let now $\rho(x, \cdot)$ be a random measure on $(a_0, +\infty)$. The measure ρ should be finite on compact sets $I \subset \mathbb{R}$. In general the measure ρ does not have to be normalizable, but it should satisfy

$$\Phi(x, a) := \rho(x, [a, \infty)) < \infty \quad \text{for all } a \in (0, \infty).$$

As we will see, $\Phi(x, a)$ has the interpretation: "Average number of cracks per unit volume of crack size $\geq a$", It is reasonable to assume that there can only be finite number of cracks larger than a in the unit volume.

We define an intensity measure $\lambda \colon \Omega \times \mathbb{R}_+ \mapsto \mathbb{R}$ by

$$\lambda(C) = \int_{\Omega} \int_{\mathbb{R}_+} \mathbf{1}_C(x, a)\, \rho(x, a)\, da dx, \quad C \subseteq \mathcal{C} \text{ is measurable}$$

with the indicator function

$$\mathbf{1}_C(x, a) = \begin{cases} 1, & (x, a) \in C \\ 0, & \text{else} \end{cases}.$$

Further, we assume a uniform distribution of cracks over Ω for the case that $\rho(x, \cdot)$ does not depend on x. Then we consider a Poisson Point Process (PPP) on the crack configuration space $\mathcal{C}$ such that

$$N(C) = \text{Number of cracks with crack configuration in } C$$

is a counting measure. From Poisson Statistics we get with the intensity measure λ:

$$P(N(C) = m) = \mathrm{e}^{-\lambda(C)} \frac{\lambda(C)^m}{m!}$$

which gives us the probability of the occurrence of m cracks in our component with shape Ω. In particular, we obtain the following probability of failure

$$\mathrm{PoF} = P(N(\mathcal{K}_{\mathrm{C}}) \geq 1) = 1 - P(N(\mathcal{K}_{\mathrm{C}}) = 0) = 1 - \mathrm{e}^{-\lambda(\mathcal{K}_{\mathrm{C}})}.$$

Now, we deliver an example for a random measure ρ which was established by Weibull in 1939 as he studied the phenomena of fracture mechanics, confer [94] and [95]. For simplicity we assume that ρ and σ_{T} do not depend on x. In this case we have a homogeneous stress state. Weibull defined ρ by

$$\rho(a) = \rho_0 a^{-\gamma},$$

where $\gamma > 1$ is a material constant. Then we obtain for the intensity measure

$$\begin{aligned}
\lambda(\mathcal{K}_{\mathrm{C}}) &= \int_{\Omega}\int_{\mathbb{R}_+} \mathbf{1}_{\mathcal{K}_{\mathrm{C}}}(x,a)\,\rho(a)\,dadx \\
&= \rho_0 \int_{\Omega}\int_{\frac{1}{\pi}\left(\frac{K_{\mathrm{Ic}}}{\sigma_{\mathrm{T}}}\right)^2}^{\infty} a^{-\gamma} dadx \\
&= \frac{\rho_0}{\gamma-1}\int_{\Omega}\left[\frac{1}{\pi}\left(\frac{K_{\mathrm{Ic}}}{\sigma_{\mathrm{T}}}\right)^2\right]^{1-\gamma} dx \\
&= \frac{\rho_0|\Omega|}{\gamma-1}\left[\frac{1}{\pi}\left(\frac{K_{\mathrm{Ic}}}{\sigma_{\mathrm{T}}}\right)^2\right]^{1-\gamma}.
\end{aligned}$$

In particular we get for the probability of failure (PoF)

$$\mathrm{PoF}(\mathcal{K}_{\mathrm{C}}) = 1 - \mathrm{e}^{-\lambda(\mathcal{K}_{\mathrm{C}})} = 1 - \exp\left(-\frac{\rho_0|\Omega|}{\gamma-1}\left[\frac{1}{\pi}\left(\frac{K_{\mathrm{Ic}}}{\sigma_{\mathrm{T}}}\right)^2\right]^{1-\gamma}\right).$$

This representation gives us the PoF for the initial crack configuration. But we are interested in a PoF as a function of the cycle number. In the next section we einclude crack dynamics into the PoF.

Dynamics of the Crack Process

We follow ideas from [7], [38] and [94]. We assume that the flow φ_n induces a flow on the space of crack configurations according to

$$\bar{\varphi}_n(c) = \bar{\varphi}_n((x,a)) = (x, \varphi(a, \sigma_{\mathrm{T}}(x))) \in \mathcal{C},\ c = (x,a) \in \mathcal{C}.$$

Physically this is based on two assumptions:

1. The crack stays where it is. It just grows in site.
2. The crack does not feel any effect from a stress change, so crack growth can be conducted with stress $\sigma(x)$. Roughly $\frac{1}{a_c - a_0} \ll \frac{\nabla\sigma_{\mathrm{T}}}{\sigma_{0,\mathrm{T}}}$ with $\sigma_{0,\mathrm{T}}$ as typical stress in the component.

We note that neglecting stress change is not necessarily conservative, in general just the opposite.

Locally, on a compact set $C \subseteq \mathcal{C}$, we have the representation of the Poisson counting measure by

$$N \restriction_C = \sum_{j=1}^{n} \delta_{c_j}$$

with Poisson random number n and random location c_j in $\mathcal{C}$. The crack growth dynamics thus transforms the counting measure N locally to

$$N \restriction_C \rightsquigarrow N_n \restriction_{\varphi_n(C)} = \sum_{j=1}^{n} \delta_{\bar{\varphi}_n(c_j)}$$

or in other words it holds

$$N(C) = N_n\left(\bar{\varphi}_n(C)\right), \quad C \subseteq \mathcal{C}.$$

If we replace C with $\bar{\varphi}_n^{-1}(C)$, then we obtain

$$N\left(\bar{\varphi}_n^{-1}(C)\right) = N_n(C) \quad \Leftrightarrow \quad N_n = \varphi_{n*} N.$$

So N_n is the image (random) counting measure of N under the flow $\bar{\varphi}$. Thus we get

$$\begin{aligned} P\left(N_n(C) = m\right) &= \mathrm{e}^{-\lambda\left(\varphi_n^{-1}(C)\right)} \frac{\lambda\left(\varphi_n^{-1}(C)\right)^m}{m!} \\ &= \mathrm{e}^{-(\varphi_{n*}\lambda)(C)} \frac{\left(\varphi_{n*}\lambda\right)(C)^m}{m!} \end{aligned}$$

for the probability of m cracks after n cycles in crack configuration C. Hence, $N_n(C)$ is a Poisson Point Process with intensity measure $\bar{\varphi}_{n*}\lambda$ on $\mathcal{C}$. Thus, we obtain the probability of failure as function of time n:

$$\begin{aligned} \mathrm{PoF}(n) &= P\left(N_n(\mathcal{K}_C) \geq 1\right) \\ &= 1 - P\left(N_n(\mathcal{K}_C) = 0\right) \\ &= 1 - \mathrm{e}^{-\bar{\varphi}_{n*}\lambda(\mathcal{K}_C)} \\ &= 1 - \mathrm{e}^{-\lambda\left(\bar{\varphi}_n^{-1}(\mathcal{K}_C)\right)}. \end{aligned}$$

We conclude

$$\bar{\varphi}_n^{-1}(c) = \bar{\varphi}_n^{-1}((x,a)) = \left(x, \varphi^{-1}\left(a, \sigma_{\mathrm{T}}(x)\right)\right) \in C$$

and we obtain

$$\varphi_n^{-1}\left(a, \sigma_{\mathrm{T}}(x)\right) = \left[a^{1-\frac{\bar{n}}{2}} - \left(1 - \frac{\bar{n}}{2}\right) A \pi^{\frac{\bar{n}}{2}} \sigma_{\mathrm{T}}^{\bar{n}}(x)\, n\right]^{\frac{1}{1-\frac{\bar{n}}{2}}}$$

for the crack length before n cycles. If we apply this to the critical set $\mathcal{K}_C$, then we obtain

$$\begin{aligned}
\bar{\varphi}_n^{-1}(\mathcal{K}_C) &= \left\{ (x, \varphi(a, \sigma_{\mathrm{T}}(x))) \in \mathcal{C} \colon a \geq \frac{1}{\pi} \left(\frac{K_{\mathrm{Ic}}}{\sigma_{\mathrm{T}}(x)} \right)^2 \right\} \\
&= \left\{ (x, a) \in \mathcal{C} \colon \varphi_n(a, \sigma_{\mathrm{T}}(x)) \geq \frac{1}{\pi} \left(\frac{K_{\mathrm{Ic}}}{\sigma_{\mathrm{T}}(x)} \right)^2 \right\} \\
&= \left\{ (x, a) \in \mathcal{C} \colon a \geq \varphi_n^{-1} \left(\frac{1}{\pi} \left(\frac{K_{\mathrm{Ic}}}{\sigma_{\mathrm{T}}(x)} \right)^2, \sigma_{\mathrm{T}}(x) \right) \right\} \\
&= \left\{ (x, a) \in \mathcal{C} \colon a \geq \left[\left(\frac{1}{\pi} \left(\frac{K_{\mathrm{Ic}}}{\sigma_{\mathrm{T}}(x)} \right)^2 \right)^{1-\frac{\bar{n}}{2}} - \left(1 - \frac{\bar{n}}{2}\right) A \pi^{\frac{\bar{n}}{2}} \sigma_{\mathrm{T}}^{\bar{n}}(x)\, n \right]^{\frac{1}{1-\frac{\bar{n}}{2}}} \right\}.
\end{aligned}$$

From this we obtain

$$\lambda_n(\mathcal{K}_{\mathcal{C}}) = \int_{\Omega} \int_{\kappa(\sigma_{\mathrm{T}}(x))}^{\infty} \bar{\rho}_x(a)\, da dx = \int_{\Omega} \Phi_x(\kappa(\sigma_{\mathrm{T}}(x)))\, dx$$

with

$$\kappa(\sigma_{\mathrm{T}}(x)) = \left[\left(\frac{1}{\pi} \left(\frac{K_{\mathrm{Ic}}}{\sigma_{\mathrm{T}}(x)} \right)^2 \right)^{1-\frac{\bar{n}}{2}} - \left(1 - \frac{\bar{n}}{2}\right) A \pi^{\frac{\bar{n}}{2}} \sigma_{\mathrm{T}}^{\bar{n}}(x)\, n \right]^{\frac{1}{1-\frac{\bar{n}}{2}}}.$$

We can understand this approach in the way of searching for the initial length of cracks which become critical after n cycles. If we want to include embrittlement, than we have to make $K_{\mathrm{Ic}} = K_{\mathrm{Ic}}(n)$ a decreasing function to model the material degeneration.

Finally, we calculate λ_n. Let

$$\Phi_x(a) = \int_a^{\infty} \bar{\rho}_x(\tilde{a})\, d\tilde{a},$$

then

$$\Phi_{n,x}(a) = \Phi_x\left(\varphi_n^{-1}(a, \sigma_{\mathrm{T}}(x))\right) = \int_{\varphi_n^{-1}(a, \sigma_{\mathrm{T}}(x))}^{\infty} \bar{\rho}_x(\tilde{a})\, d\tilde{a}$$

and

$$\bar{\rho}_{n,x}(a) = -\frac{d}{da} \Phi_{n,x}(a) = \bar{\rho}_x\left(\varphi_n^{-1}(a, \sigma_{\mathrm{T}}(x))\right) \frac{d}{da} \varphi_n^{-1}(a, \sigma_{\mathrm{T}}(x)).$$

If we use the two observations, then we finally get

$$\lambda_n(C) = \int_{\Omega} \int_{\mathbb{R}_+} \mathbf{1}_C(x, a)\, \bar{\rho}_{n,x}(a)\, da dx.$$

Now, we are able to calculate the probability of failure after n cycles. In the next section we want to investigate the effect of an inspection.

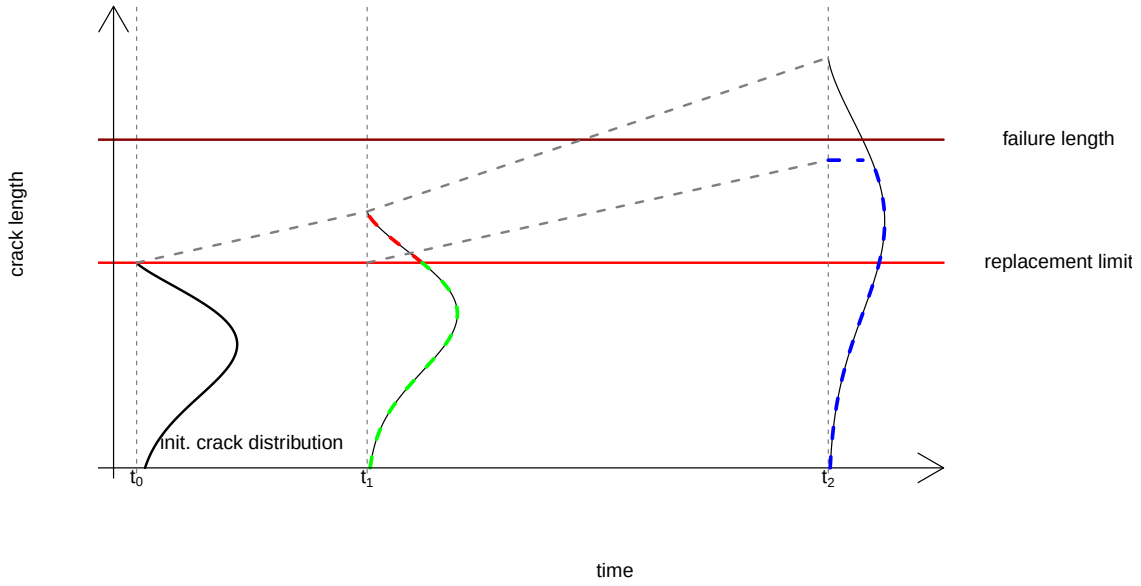


Figure 6.5.: The black line refers to the initial crack length distribution at t_0. The red and green dashed line refers to crack distribution at t_1. The part will be replaced during an inspection at t_1, if the crack length is in the red dashed section. The blue dashed line at t_2 refers to the crack length distribution, if there was an inspection at t_1 without an indication. The black line is the corresponding distribution, if there was no inspection at t_1.

The Effect of Inspection

We assume that our inspection procedure can detect a crack with length of a with a certain probability of detection $\mathrm{PoD}(a)$. In the extreme case, when all cracks $\geq a_0$ are detected and of size below a_0, $\mathrm{PoD}(a)$ is the step function. We consider this case for simplicity. In addition we assume for simplicity that the component will always be exchanged, if a crack is detected. $\mathrm{PoD}(a)$ has the properties of a distribution function in the crack length a.
We first calculate the probability that after n load cycles a crack is detected. To reach this aim, let $\bar{\rho}_{x,n}(a)$ be the original density and let $\rho^D_{x,n}(a) = \mathrm{PoD}(a)\,\bar{\rho}_{x,n}(a)$. Accordingly, we define

$$\lambda^D_n(C) = \int_\Omega \int_{\mathbb{R}_+} \mathbf{1}_C(x,a)\,\rho^D_{x,n}(a)\,dadx.$$

Let N^D_n is the Poisson point process associated to the intensity measure λ^D_n. Then N^D_n is the counting measure of detected cracks. The probability for a premature exchange

of the component after n load cycles is thus

$$\begin{aligned}\mathrm{PoE}(n) &= P\left(N_n^D(\mathcal{C}) \geq 1\right)\\ &= 1 - P\left(N_n^D(\mathcal{C}) = 0\right)\\ &= 1 - \mathrm{e}^{-\lambda_n^D(\mathcal{C})}, \quad \mathcal{C} = \Omega \times \mathbb{R}_+.\end{aligned}$$

If the component is exchanged, start with λ_0 for the new component. If the component is not exchanged, then replace $\bar{\rho}_{0,x}(a)$ with $\bar{\rho}_{n,x}^N(a) = (1 - \mathrm{PoD}(a))\,\bar{\rho}_{n,x}(a)$ with n the number of load cycles since last inspection and the overall load cycles $N \geq n$. Then we start recursively from the beginning. In detail we get for our intensity measure of failure

$$\begin{aligned}\lambda_N^{F,n}(\mathcal{C}) &= \int_\Omega \int_{\mathbb{R}_+} \mathbf{1}_C(x,a)\,\bar{\rho}_{n,x}^N(a)\,dadx\\ &= \int_\Omega \int_{\mathbb{R}_+} \mathbf{1}_C(x,a)\,(1 - \mathrm{PoD}(a))\,\bar{\rho}_{n,x}(a)\,dadx\\ &= \int_\Omega \int_{a_N^F(x)}^{a_n^D(x)} \bar{\rho}_x(a)\,dadx\end{aligned}$$

with the critical detection length

$$a_n^D(x) = \varphi_n^{-1}(a_0, \sigma_{\mathrm{T}}(x))$$

and the critical failure length

$$a_N^{F,n}(x) = \min\left(a_n^D(x), \left[\left(\frac{1}{\pi}\left(\frac{K_{\mathrm{Ic}}}{\sigma_{\mathrm{T}}(x)}\right)^2\right)^{1-\frac{\bar{n}}{2}} - \left(1 - \frac{\bar{n}}{2}\right) A\pi^{\frac{\bar{n}}{2}}\sigma_{\mathrm{T}}^{\bar{n}}(x)\,n\right]^{\frac{1}{1-\frac{\bar{n}}{2}}}\right).$$

This leads to the following formula for the probability of a failure after an inspection at n and N overall load cycles

$$\mathrm{PoF}(N,n) = 1 - \mathrm{e}^{-\lambda_N^{F,n}(\mathcal{C})}. \tag{6.18}$$

Here, we see very clearly the effect of inspection: It reduces the integration area in the second integral from $\left(a_N^F, \infty\right)$ to $\left(a_N^F, a_n^D\right)$. This property is shown in Figure 6.5. Analogously, we get the intensity measure $\lambda_N^{D,n}$ for the PoE after an inspection at n load cycles by

$$\lambda_N^{D,n}(\mathcal{C}) = \int_\Omega \int_{a_N^D(x)}^{a_n^D(x)} \bar{\rho}_x(a)\,dadx$$

with

$$a_n^D(x) = \varphi_n^{-1}(a_0, \sigma_{\mathrm{T}}(x)) \quad \text{and} \quad a_N^D(x) = \varphi_N^{-1}(a_0, \sigma_{\mathrm{T}}(x)).$$

The PoE is then given by

$$\mathrm{PoE}(N,n) = 1 - \mathrm{e}^{-\lambda_N^{D,n}(\mathcal{C})}. \tag{6.19}$$

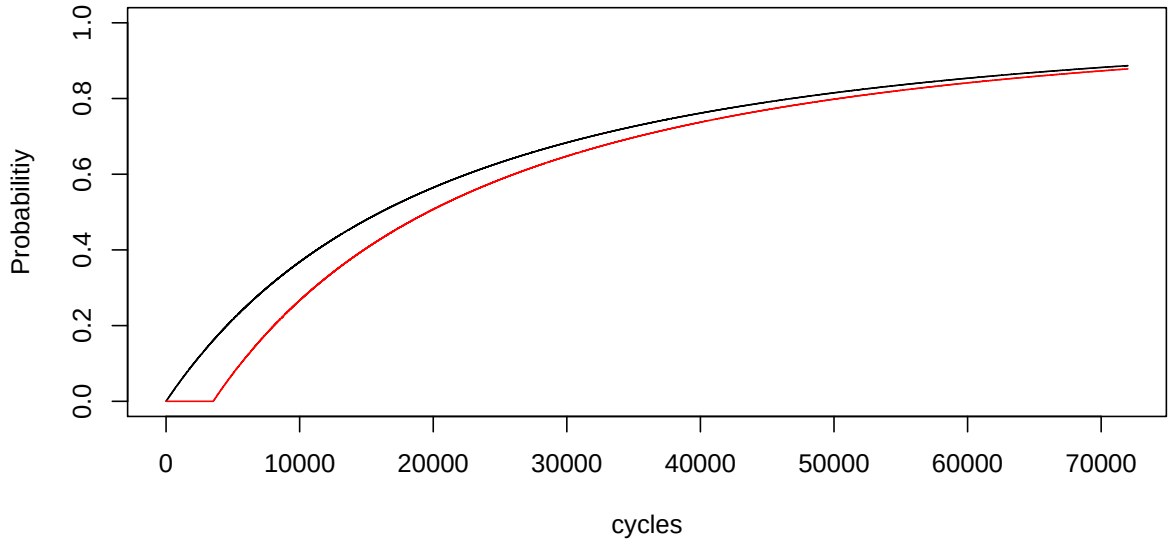


Figure 6.6.: The black line represents the PoE for a new part and the red line represents the associated PoF.

We can summarize that the PoF and PoE depends only on the overall load cycles N and the load cycles n at the last inspection. In particular, the PoE and PoF after an inspection n are not influenced by inspection prior to n in the case that no exchange was performed. Therefore, the Markov property holds. We can see this effect in Figure 6.5. Further, we see in Figure 6.6 an exemplary trend for the PoE (black line) and PoF (red line) for a new part.

6.3. The Inspection Model

We start with the description of our inspection model in terms of a partially observable Markov Decision Process. In the next sections we define the different elements of the POMDP and how they are connected to the gas turbine maintenance/inspection model.

The Time

As first step we divided the reviewed time interval $[0, T_{\text{end}}]$ of our maintenance scheduling problem into $T \in \mathbb{N}$ equal distributed stages, e.g. we get a set $\{t_0 = 0, t_1, \ldots, t_T = T_{\text{end}}\}$ with $t_{i+1} = t_i = \text{const}$ for all $i = 1, \ldots, T-1$. To simplify the notation we mention only the stage number $t = 1, \ldots, T$ instead of the real time points t_i. This is the same approach as in Section 5.3.

The State Space

We assume that our gas turbine consists of m components. Further, we assume that every component can be inspected and there is one inspection criterion which is checked. In this case we will check for a crack of a certain length. The state x of the i-th component is given through a 3 dimensional vector. In particular we have

$$x_i = (x_{\mathrm{A}i}, x_{\mathrm{I}i}, x_{\mathrm{C}i}) \in \{-1, d_1, \ldots, d_n = d_{\max},\} \times \{0, d_1, \ldots, d_{n-1},\} \times \{-1, 0\} =: \mathcal{S}_i$$

where x_{A} is either the age of the component or it indicates the failure state $x_{\mathrm{A}} = -1$, x_{I} is the age at the last inspection and x_{C} indicates that the inspection criterion is fulfilled $x_{\mathrm{C}} = 0$ or not $x_{\mathrm{C}} = -1$. Like in Section 5.3 the discretization of the age of the component depends on the time discretization and the maximal allowable age of the component. The complete state x of the gas turbine is defined by

$$\mathcal{S} := \mathcal{S}_1 \times \ldots \times \mathcal{S}_m.$$

We remark that the size of the state space growth faster than in the advanced replacement model, because the size $|\mathcal{S}_i|$ is much bigger. Further, we remark that in this model we measure the age in equivalent starts (ES) which fits better to the cyclic damage mechanism.

The Action Space

The action space $\mathcal{A}_i$ for the i-th component is defined by

$$\mathcal{A}_i := \{0, 1\}^2.$$

Further, we split $a \in \mathcal{A}_i$ into $a = (a_{\mathrm{R}}, a_{\mathrm{I}})$ where a_{R} indicates that we replace ($a_{\mathrm{R}} = 1$) the component or not ($a_{\mathrm{R}} = 0$). The second component a_{I} represents the service type of inspection with $a_{\mathrm{I}} = 1$ inspect the component and $a_{\mathrm{I}} = 0$ do not execute an inspection. Since, it is only possible to do nothing, replace or inspect the component, we introduce the constraint

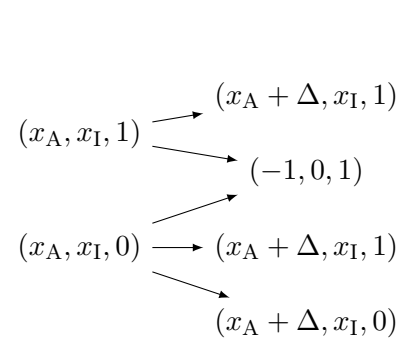
$$a_{\mathrm{R}} + a_{\mathrm{I}} \leq 1.$$

If a component is in the failed state $x_{\mathrm{A}} = -1$, we can only execute the service action replacement, e.g. $a = (1, 0)$. The complete action state space $\mathcal{A}$ for multi component system is defined by

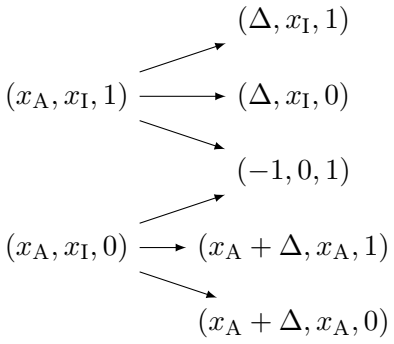
$$\mathcal{A} := \mathcal{A}_1 \times \ldots \times \mathcal{A}_m.$$

The State Dynamics

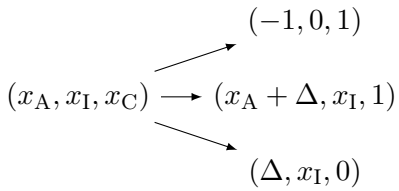
The random variable ω_t is split into two components $\omega_t = (\omega_{\mathrm{F}}, \omega_{\mathrm{I}}) \in \{0, 1\}^2$. The first component ω_{F} indicates a failure, e.g. $\omega_{\mathrm{F}} = 1$ and the second component ω_{I} indicates



(a) Transition graph for $a = (0, 0)$



(b) Transition graph for $a = (1, 0)$



(c) Transition graph for $a = (0, 1)$

Figure 6.7.: The transition graphs for all actions a.

that the inspection criterion is fulfilled $\omega_I = 0$ or not $\omega_I = 1$. Further, we have the constraint $\omega_I \geq \omega_F$ to ensure that a failed component does not fulfill the inspection criterion. This follows directly, because the component fails due to the existence of a critical crack and this crack length is longer than the crack length for the inspection criterion. The two events are not independent. Then the state dynamics function $g_i(x_t, a_t, \omega_t)$ of the state x_t of the i-th component is defined by

$$g_i(x_t, a_t, \omega_t) = \begin{cases} x_{t+1} = (x_A + \Delta c_t, x_I, \omega_I), & a_t = (0,0),\ \omega_t = (0, \omega_I) \\ x_{t+1} = (\Delta c_t, 0, \omega_I), & a_t = (1,0),\ \omega_t = (0, \omega_I) \\ x_{t+1} = (x_A + \Delta c_t, x_A, \omega_I), & a_t = (0,1),\ \omega_t = (0, \omega_I),\ x_C = 0 \\ x_{t+1} = (\Delta c_t, 0, \omega_I), & a_t = (0,1),\ \omega_t = (0, \omega_I),\ x_C = 1 \\ x_{t+1} = (-1, 0, -1), & \omega_t = (1,1) \end{cases}.$$

The state dynamics are illustrated in Figure 6.7 for every possible action $a \in \mathcal{A}_i$. The complete state dynamics for all m components are given by the following function

$$g(x_t, a_t, \omega_t) = (g_1(x_{t1}, a_{t1}, \omega_{t1}), \ldots, g_m(x_{tm}, a_{tm}, \omega_{tm})).$$

The Transition Probabilities

Now, we use the failure and exchange probabilities from Section 6.2.1, the probabilistic crack growth model, to create the transition probabilities. As described in Section 6.2.1,

we assume that the probability of detection is a step function, and we assume that we replace the component, if we find a crack during an inspection. The critical crack length for a premature replacement is equal to the detection length. Therefore, we take the three probabilities PoF, PoE and PoC according to equations (6.18) and (6.19) into account. In detail we have

$$\mathrm{PoF}\left(x_{\mathrm{A}}, x_{\mathrm{I}}\right) \quad \text{and} \quad \mathrm{PoE}\left(x_{\mathrm{A}}, x_{\mathrm{I}}\right).$$

Then the conditional transition probabilities $Pr\left(x_{t+1}|x_t, a_t\right)$ for the different service actions are defined by

Do nothing $a_t = (0, 0)$

- No failure and inspection criterion fulfilled:

  $$Pr\left(x_{t+1}|x_t, a_t\right) = 1 - \mathrm{PoE}\left(x_{\mathrm{A}t+1}, x_{\mathrm{A}t}\right)$$

  with $x_{\mathrm{A}t}, x_{\mathrm{A}t+1}, x_{\mathrm{C}t}, x_{\mathrm{C}t+1} > 0$. This case is equivalent to there being no crack at t and $t+1$.

- Failure:

  $$Pr\left(x_{t+1}|x_t, a_t\right) = \mathrm{PoF}\left(x_{\mathrm{A}t+1}, x_{\mathrm{A}t}\right)$$

  with $x_{\mathrm{A}t}, x_{\mathrm{C}t} > 0,\ x_{\mathrm{A}t+1}, x_{\mathrm{C}t+1} = -1$.

- No failure and inspection criterion not fulfilled:

  $$\begin{aligned} Pr\left(x_{t+1}|x_t, a_t\right) &= 1 - \left(1 - \mathrm{PoE}\left(x_{\mathrm{A}t+1}, x_{\mathrm{A}t}\right) + \mathrm{PoF}\left(x_{\mathrm{A}t+1}, x_{\mathrm{A}t}\right)\right) \\ &= \mathrm{PoE}\left(x_{\mathrm{A}t+1}, x_{\mathrm{A}t}\right) - \mathrm{PoF}\left(x_{\mathrm{A}t+1}, x_{\mathrm{A}t}\right) \end{aligned}$$

  with $x_{\mathrm{A}t}, x_{\mathrm{A}t+1}, x_{\mathrm{C}t} > 0,\ x_{\mathrm{C}t+1} = -1$.

- No failure, but inspection criterion keeps unfulfilled:

  $$Pr\left(x_{t+1}|x_t, a_t\right) = 1 - \mathrm{PoF}\left(x_{\mathrm{A}t+1}, x_{\mathrm{I}t}\right)$$

  with $x_{\mathrm{A}t}, x_{\mathrm{A}t+1} > 0,\ x_{\mathrm{C}t+1}, x_{\mathrm{C}t} = -1$.

- Failure, but inspection criterion was unfulfilled:

  $$Pr\left(x_{t+1}|x_t, a_t\right) = \mathrm{PoF}\left(x_{\mathrm{A}t+1}, x_{\mathrm{I}t}\right)$$

  with $x_{\mathrm{A}t} > 0,\ x_{\mathrm{C}t+1}, x_{\mathrm{C}t}, x_{\mathrm{A}t+1} = -1$.

Replacement $a_t = (1, 0)$

- No Failure and inspection criterion fulfilled:

  $$Pr\left(x_{t+1}|x_t, a_t\right) = 1 - \mathrm{PoE}\left(x_{\mathrm{A}t+1}, 0\right),$$

  with $x_{\mathrm{A}t+1} = \Delta c_t,\ x_{\mathrm{I}t+1} = 0,\ x_{\mathrm{C}t+1} = 0$.

- Failure:

$$Pr\left(x_{t+1}|x_t, a_t\right) = \mathrm{PoF}\left(x_{\mathrm{A}t+1}, 0\right),$$

with $x_{\mathrm{A}t+1} = -1,\ x_{\mathrm{I}t+1} = 0,\ x_{\mathrm{C}t+1} = -1$.

- No Failure and inspection criterion not fulfilled:

$$\begin{aligned} Pr\left(x_{t+1}|x_t, a_t\right) &= 1 - \left(1 - \mathrm{PoE}\left(x_{\mathrm{A}t+1}, 0\right) + \mathrm{PoF}\left(x_{\mathrm{A}t+1}, 0\right)\right) \\ &= \mathrm{PoE}\left(x_{\mathrm{A}t+1}, 0\right) - \mathrm{PoF}\left(x_{\mathrm{A}t+1}, 0\right) \end{aligned}$$

with $x_{\mathrm{A}t+1} = \Delta c_t,\ x_{\mathrm{I}t+1} = 0,\ x_{\mathrm{C}t+1} = -1$.

- Remark: In this case we need no distinction of the state at x_t, because the replacement set the component back into the "as new" state.

Inspection $a_t = (0, 1)$

- No failure and inspection criterion fulfilled:

$$Pr\left(x_{t+1}|x_t, a_t\right) = 1 - \mathrm{PoE}\left(x_{\mathrm{A}t+1}, x_{\mathrm{A}t}\right)$$

with $x_{\mathrm{A}t}, x_{\mathrm{A}t+1}, x_{\mathrm{C}t}, x_{\mathrm{C}t+1} > 0,\ x_{\mathrm{I}t+1} = x_{\mathrm{A}t}$.

- No failure and inspection criterion not fulfilled:

$$\begin{aligned} Pr\left(x_{t+1}|x_t, a_t\right) &= 1 - \left(1 - \mathrm{PoE}\left(x_{\mathrm{A}t+1}, x_{\mathrm{A}t}\right) + \mathrm{PoF}\left(x_{\mathrm{A}t+1}, x_{\mathrm{A}t}\right)\right) \\ &= \mathrm{PoE}\left(x_{\mathrm{A}t+1}, x_{\mathrm{A}t}\right) - \mathrm{PoF}\left(x_{\mathrm{A}t+1}, x_{\mathrm{A}t}\right) \end{aligned}$$

with $x_{\mathrm{A}t}, x_{\mathrm{A}t+1}, x_{\mathrm{C}t} > 0,\ x_{\mathrm{I}t+1} = x_{\mathrm{A}t},\ x_{\mathrm{C}t+1} = -1$.

- Failure:

$$Pr\left(x_{t+1}|x_t, a_t\right) = \mathrm{PoF}\left(x_{\mathrm{A}t+1}, x_{\mathrm{A}t}\right)$$

with $x_{\mathrm{A}t}, x_{\mathrm{C}t} > 0,\ x_{\mathrm{I}t+1} = 0,\ x_{\mathrm{A}t+1}, x_{\mathrm{C}t+1} = -1$.

- No failure, no crack and premature replacement:

$$Pr\left(x_{t+1}|x_t, a_t\right) = 1 - \mathrm{PoE}\left(x_{\mathrm{A}t+1}, 0\right)$$

with $x_{\mathrm{A}t}, x_{\mathrm{C}t+1} > 0,\ x_{\mathrm{I}t+1} = 0,\ x_{\mathrm{A}t+1} = \Delta c_t,\ x_{\mathrm{C}t} = -1$.

- No failure, crack and premature replacement:

$$\begin{aligned} Pr\left(x_{t+1}|x_t, a_t\right) &= 1 - \left(1 - \mathrm{PoE}\left(x_{\mathrm{A}t+1}, 0\right) + \mathrm{PoF}\left(x_{\mathrm{A}t+1}, 0\right)\right) \\ &= \mathrm{PoE}\left(x_{\mathrm{A}t+1}, 0\right) - \mathrm{PoF}\left(x_{\mathrm{A}t+1}, 0\right) \end{aligned}$$

with $x_{\mathrm{A}t}, x_{\mathrm{C}t+1} > 0,\ x_{\mathrm{I}t+1} = 0,\ x_{\mathrm{A}t+1} = \Delta c_t,\ x_{\mathrm{C}t} = -1$.

- Failure and premature replacement:

$$Pr\left(x_{t+1}|x_t, a_t\right) = \mathrm{PoE}\left(x_{\mathrm{A}t+1}, 0\right)$$

  with $x_{\mathrm{A}t} > 0,\ x_{\mathrm{I}t+1} = 0,\ x_{\mathrm{A}t+1} = \Delta c_t,\ x_{\mathrm{C}t+1}, x_{\mathrm{C}t} = -1.$

- Remark: The last three cases for premature replacement are same as in the replacement case.

As in the last chapter, the over all transition function Pr for the gas turbine state with m components is defined by

$$Pr\left(x_{t+1}|x_t, a_t\right) := \prod_{i=1}^{m} Pr\left(x_{t+1i}|x_{ti}, a_{ti}\right).$$

The Reward Function

As in our replacements models, the inspection model's reward function G_t represents the revenue of the gas turbine operator in the time step t. The reward function is divided into three parts. The first part gives information about the revenue I_t per time step. The second part summarize the maintenance cost in dependency of the service action a_t and the state x_t. The service cost is split into four parts. We have equipment cost $C_{\mathrm{ME}ti} > 0$, assembly cost $C_{\mathrm{MA}ti} > 0$, inspection cost $C_{\mathrm{MI}ti} > 0$ and a penalty fee $C_{\mathrm{MP}ti} > 0$ for a premature replacement during an inspection. To penalize a premature replacement we claim $C_{\mathrm{MP}ti} > C_{\mathrm{ME}ti}$. The last part represents the failure cost $C_{\mathrm{R}ti} \geq 0$ per component. In summary we have

$$\begin{aligned} G_t\left(x, a\right) = I_t &+ \sum_{i=1}^{m} C_{\mathrm{R}i} \min\left(0, x_{\mathrm{A}i}\right) \\ &- \sum_{i=1}^{m} C_{\mathrm{ME}ti} a_{\mathrm{R}i} - \left(C_{\mathrm{MI}ti} - C_{\mathrm{MP}ti} \min\left(0, x_{\mathrm{C}i}\right)\right) a_{\mathrm{I}i} \\ &- \sum_{i=1}^{m} \max_i \left(\max\left(a_{\mathrm{R}i}, a_{\mathrm{I}i}\right) \sum_{j=1}^{i} C_{\mathrm{MA}tj} \right). \end{aligned}$$

In this model we assume as before, that if we want to disassemble component i, then we have to dismantle all components j with $j < i$. The term $C_{\mathrm{MI}ti} - C_{\mathrm{MP}ti} \min\left(0, x_{\mathrm{C}i}\right)$ summarizes the cost for an inspection and includes the penalty for a premature replacement, if needed.

The Observation Space

According to Definition 6.1 we need to define a observation space $\mathcal{O}$ for inspection model as POMDP. The observation space $\mathcal{O}$ is a subset of our state space $\mathcal{S}$. In particular we

have

$$\mathcal{O} := \{x \in \mathcal{S} \colon x_{\mathrm{C}} = 0\} \cup \{(-1, 0, -1)\}\,.$$

This choice is reasonable, because we cannot see the cracks before we perform an inspection for various parts. Further, a detected crack during an inspection leads to a premature replacement. Therefore, we will never leave a part with detected crack inside the gas turbine. But the age x_{A} and the age x_{I} at the last inspection is traceable during operation. Also, we can detect or observe a failure of a component at every time.

The Observation Function

As last missing part of the POMDP model, we need to define the observation function $Pr\left(o_t | x_t, a_{t-1}\right)$ which gives us a probability to make the observation $o_t \in \mathcal{O}$, if we performed action a_{t-1} and we are in the resulting state x_t. In general, we have the following three cases for a single component

Do nothing $a_{t-1} = (0, 0)$

$$Pr\left(o_t | x_t, a_{t-1}\right) := \begin{cases} 1, & o_{\mathrm{A}} = x_{\mathrm{A}},\ o_{\mathrm{I}} = x_{\mathrm{I}},\ x_{\mathrm{A}} - x_{\mathrm{I}} > \Delta c_t \\ 1, & x_{\mathrm{A}} = o_{\mathrm{A}} = -1,\ x_{\mathrm{I}} = o_{\mathrm{I}} = 0,\ x_{\mathrm{C}} = -1 \\ 0, & \text{else} \end{cases}$$

Replacement $a_{t-1} = (1, 0)$

$$Pr\left(o_t | x_t, a_{t-1}\right) := \begin{cases} 1, & o_{\mathrm{A}} = x_{\mathrm{A}} = \Delta c_t,\ o_{\mathrm{I}} = x_{\mathrm{I}} = 0 \\ 1, & x_{\mathrm{A}} = o_{\mathrm{A}} = -1,\ x_{\mathrm{I}} = o_{\mathrm{I}} = 0,\ x_{\mathrm{C}} = -1 \\ 0, & \text{else} \end{cases}$$

Inspection $a_{t-1} = (0, 1)$

$$Pr\left(o_t | x_t, a_{t-1}\right) := \begin{cases} 1, & o_{\mathrm{A}} = x_{\mathrm{A}},\ o_{\mathrm{I}} = x_{\mathrm{I}} > 0,\ x_{\mathrm{A}} - x_{\mathrm{I}} = \Delta c_t \\ 1, & o_{\mathrm{A}} = x_{\mathrm{A}} = \Delta c_t,\ o_{\mathrm{I}} = x_{\mathrm{I}} = 0 \\ 1, & x_{\mathrm{A}} = o_{\mathrm{A}} = -1,\ x_{\mathrm{I}} = o_{\mathrm{I}} = 0,\ x_{\mathrm{C}} = -1 \\ 0, & \text{else} \end{cases}$$

This mapping follows from the definition of the observation space $\mathcal{O}$ and the stated dynamics g, because the single uncertainty in the state is the fulfillment of the inspection criterion which is represented by x_{C}. Further, we can only reach distinct states after a service action a.

6.4. Numerical Analysis

In this section, we present a numerical analysis of the inspection model in terms of a POMDP. In the first part, we describe the numerical implementation. In the second part, we show numerical results for example models and compare the results and runtimes of Algorithm 6.3 and Algorithm 6.5.

But first, we describe the model which we analyze. We consider a gas turbine which consists of one component only. We consider only one component to reduce the computational effort. We allow the three service actions "do nothing" $a = (0, 0)$, "replacement" $a = (1, 0)$ and "inspection" $a = (0, 1)$ from Section 6.3. We take a time duration of 12 years and 2400 operating hours per year into account. Further, we assume 25 starts per month with 8 operating hours per start. We summarize all important model parameters for the inspection model in the following list:

Revenue function parameters:

- Failure cost $C_{\mathrm{R}} = 50.0\text{Mio. Euro}$
- Equipment cost $C_{\mathrm{ME}} = 3.0\text{Mio. Euro}$
- Assembly cost $C_{\mathrm{MA}} = 0.15\text{Mio. Euro}$
- Inspection cost $C_{\mathrm{MI}} = 0.6\text{Mio. Euro}$
- Inspection penalty fee $C_{\mathrm{MP}} = 6.3\text{Mio. Euro}$
- Revenue $I = 4\frac{\text{Euro}}{\text{MWh}} \cdot 300\text{MW} \cdot 8\frac{\text{h}}{\text{start}} \cdot 25\frac{\text{start}}{\text{month}} \cdot \Delta t = 240000\frac{\text{Euro}}{\text{month}} \cdot \Delta t$
- Nominal discount factor $i_{\mathrm{no}} = 0.05$

Crack model parameters:

- Material constants $A = 3.86 \cdot 10^{-10}$, $\gamma = 1.05$, $\bar{n} = 2.25$
- Detection length $a_{\mathrm{det}} = 0.81\text{cm}$
- Failure length $a_{\mathrm{crit}} = 4.06\text{cm}$

For simplicity we assume in the crack model that all model parameters do not depend on the location x of the component. Further, we assume a Weibull model like in the example in Section 6.2.1. The material parameters are comparable with the data from [61]. The financial data are based on the assumption from Chapters 4 and 5.

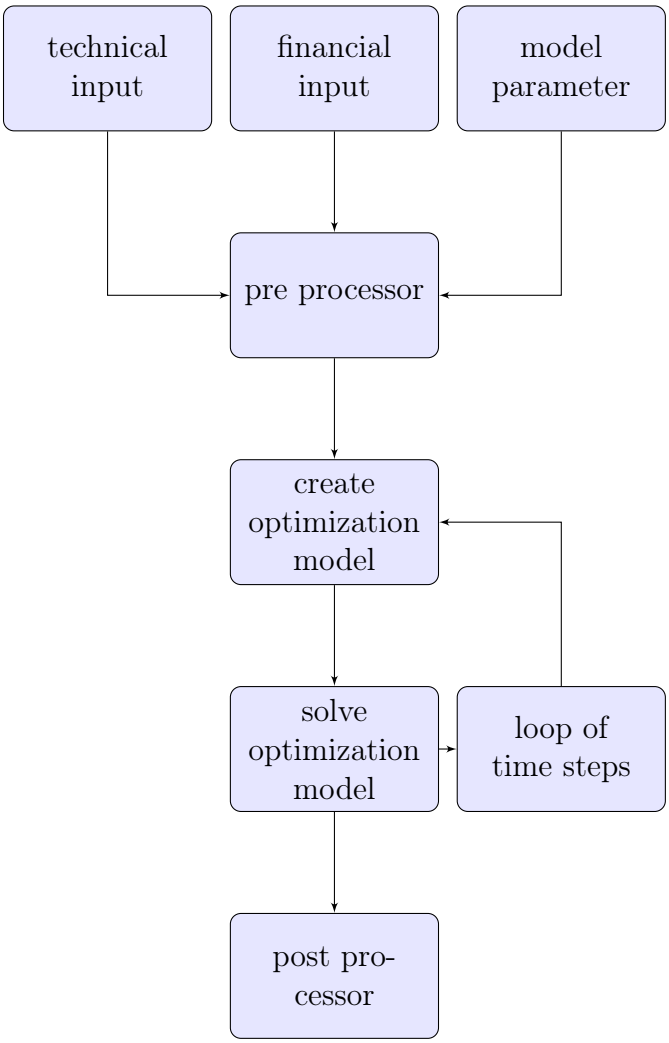


Figure 6.8.: Flow chart of the inspection model implementation.

6.4.1. Numerical Implementation

Basically, we implemented the two Algorithms 6.3 and 6.5 in R, [75], and used gurobi, [46], as LP solver. The used software versions and hardware configuration is the same as in Chapter 5. The implementation concept is similar two the one of the replacement model, compare Figure 4.10 and Figure 6.8. The main difference is that we have to loop over the time steps and solve many optimization problems. At this point we parallelize the execution of the the optimization problems.

In Algorithm 6.3 we parallelize the prune operation over the three different action $a \in \mathcal{A}$ to save computational time. We get the best set of α-vectors for every action a and can reduce it then to $\tilde{\mathcal{V}}_t$ finally. Therefore, we can use 3 cpu cores simultaneously since $|\mathcal{A}| = 3$.

Algorithm 6.7 Back-Projection (BL): Calculates $\bar{\alpha}_t$ for a given belief state b.

function BL(b, $\mathcal{V}_{t+1}$, R_a)
  for $a \in \mathcal{A}$ **do** ▷ This loop will be used for a parallel implementation.
    $\bar{\alpha}_t^a \leftarrow R_a$
    for $o \in \mathcal{O}$ **do** ▷ This loop will be used for a parallel implementation.
      $\bar{\alpha}_t^a \leftarrow \bar{\alpha}_t^a + \arg\max_{\alpha_{t+1} \in \mathcal{V}_{t+1}} \left(\mathrm{e}^{-i_{\mathrm{no}}} \, Pr\left(o|a,b\right) \alpha_{t+1}^{\mathrm{T}} b \right)$
    end for
  end for
  $\bar{\alpha}_t \leftarrow \arg\max_{\bar{\alpha}_t^a} \left(\bar{\alpha}_t^a\right)^{\mathrm{T}} b$
  return $\bar{\alpha}_t$
end function

In Algorithm 6.5 we use a parallel implementation of the back projection algorithm which calculates $\bar{\alpha}$ or in particular it finds the best α-vector from the set $P\left(\tilde{\mathcal{V}}_{t+1}\right)$ for a given belief state $b \in \Pi\left(\mathcal{S}\right)$. For the definition of the back projection algorithm confer Algorithm 6.7. This algorithm parallelize the loop over the different actions $a \in \mathcal{A}$. Further, for every action $a \in \mathcal{A}$ we split the the observation space $\mathcal{O}$ into #cpu core subsets $\tilde{\mathcal{O}}_i$. By this measure, we can use more than one cpu core for every action $a \in \mathcal{A}$ to find $\bar{\alpha}$. In our implementation approach we use 14 cpu cores.

In addition gurobi takes advantage of a multi cpu core system in both algorithms. The LP-solver gurobi runs a different algorithm / solution strategy on every cpu core and takes the solution from the fastest one, confer [44]. In our case it uses 16 cpu cores. This leads to 16 different strategies.

In the α-min-Algorithm 6.6 we choose the following parameters to bound the iteration steps. First, we define $\tilde{N}$ as number of the unique α-vectors which are created due to the back projection of the $|\mathcal{S}|$ unity vectors $e_i \in \Pi\left(\mathcal{S}\right)$ where the i-th component takes the value 1. We can calculate these back projected α-vectors very fast, because it can be done simultaneously. Then we take $N = \tilde{N} + 5 \leq |\mathcal{S}| + 5$ as maximal number of α-vectors per time step. For the maximal number $n_{\max}$ of iteration steps to approximate the convex polyhedron C_t by $\tilde{C}_t^n$ we choose $n_{\max} = 10$.

6.4.2. Numerical Results

To measure the error of the α-min-Algorithm we can use the error gap from Lemma 6.6 like presented in Table 6.1. Further, we create randomly 10^4 belief states and use the absolute error measure according to equation (6.11), e.g.

$$val_{\mathrm{abs}}\left(\mathcal{V}_t, \tilde{\mathcal{V}}_t, b\right) := \left| \max_{\alpha \in \mathcal{V}_t} \alpha^{\mathrm{T}} b - \max_{\alpha \in \tilde{\mathcal{V}}_t} \alpha^{\mathrm{T}} b \right|, \tag{6.20}$$

to calculate the error. The results for time step $t = 1$ are presented in the histograms in Figure 6.11. We see that the approximation fits in all four cases very well. We compare

T	Δt	$\lvert\mathcal{S}\rvert$	runtime [mean time per stage] in sec		#α-vectors		gap
			IP	α-min	IP	α-min	err
40	3	2353	1208.9 [30.22]	61334.3 [98]	117	25828	121.504
20	6	601	86.7 [4.34]	4737.0 [236.85]	57	3481	58.7
10	12	157	8.8 [0.88]	1377.8 [137.78]	27	607	27.5
5	24	43	1.1 [0.23]	435.2 [87.04]	12	105	11.9

Table 6.1.: Different result parameters for the IP and α-min-Algorithm. The runtime in brackets [·] is the average runtime per time stage. #α is the number of α-vectors over all time stages t. Δt is given in months.

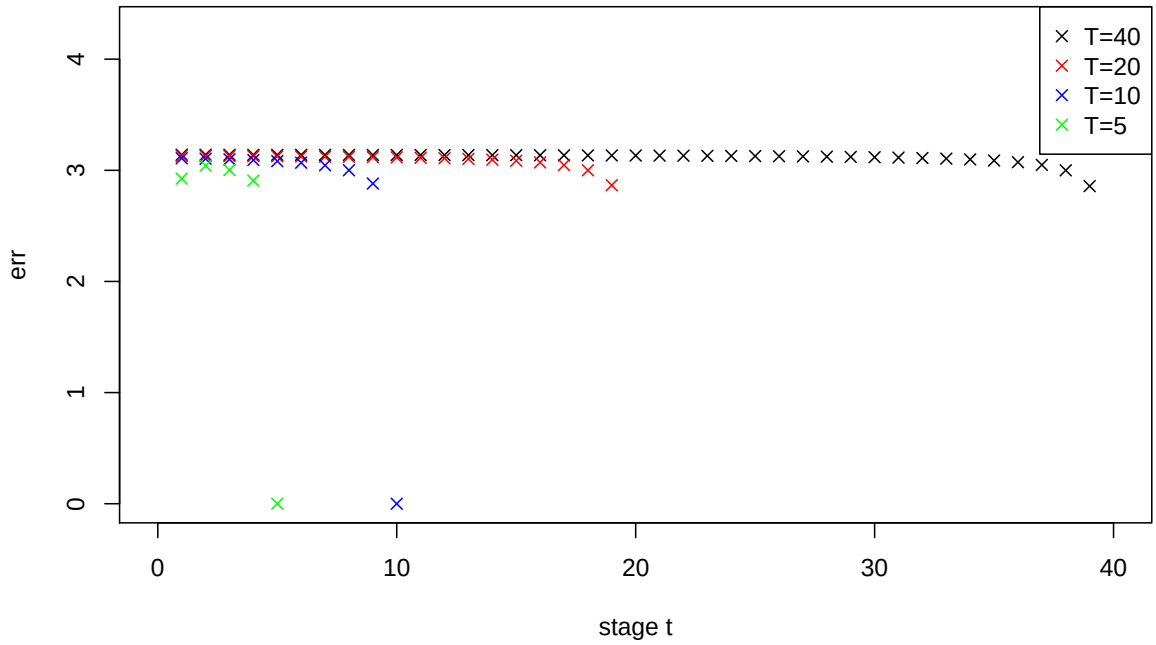


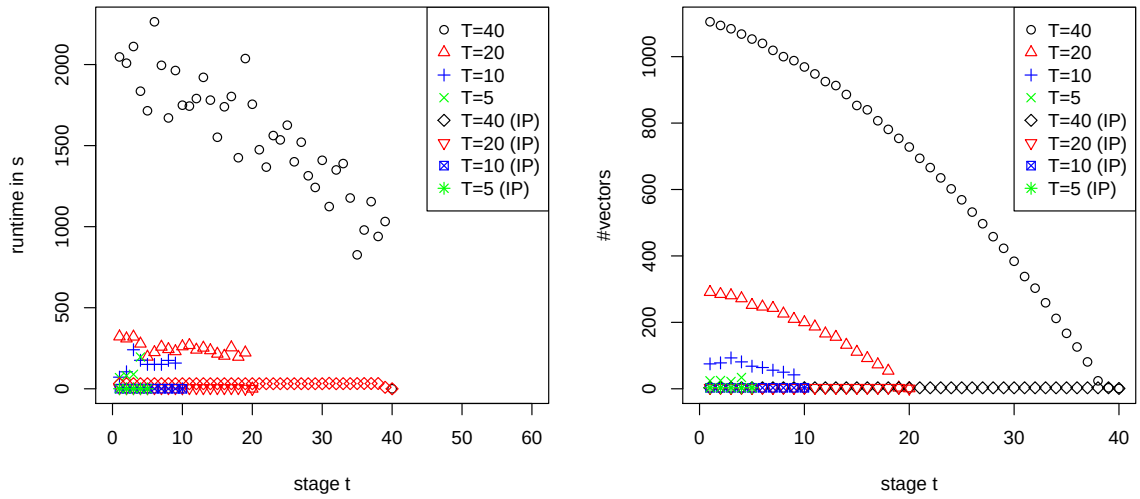
Figure 6.9.: The error gap err_t from Algorithm 6.6 over the time.

the results of the IP and α-min-Algorithm with a second error measure according to equation (5.18), e.g. we have

$$val_{\text{rel}}\left(\mathcal{V}_t, \tilde{\mathcal{V}}_t, b\right) := \left|1 - \frac{\max_{\alpha\in\tilde{\mathcal{V}}_t} \alpha^{\mathrm{T}} b}{\max_{\alpha\in\mathcal{V}_t} \alpha^{\mathrm{T}} b}\right|. \tag{6.21}$$

Therefore, we use the 10^4 randomly created belief states to create the histograms in Figure 6.12 to compare the approximate results with the real results from the IP-Algorithm. In this case the error is very small. We can conclude that, the created solution from the α-min-Algorithm approximates the real solution very well. In Figure 6.9 we present the error gap err_t over the different time steps. We can conclude that the error does not change significantly over the time steps.

However, the runtime of the α-min-Algorithm is even worse compared to the exact IP-Algorithm. As we see in Table 6.1 the runtimes of the α-min-Algorithm are much

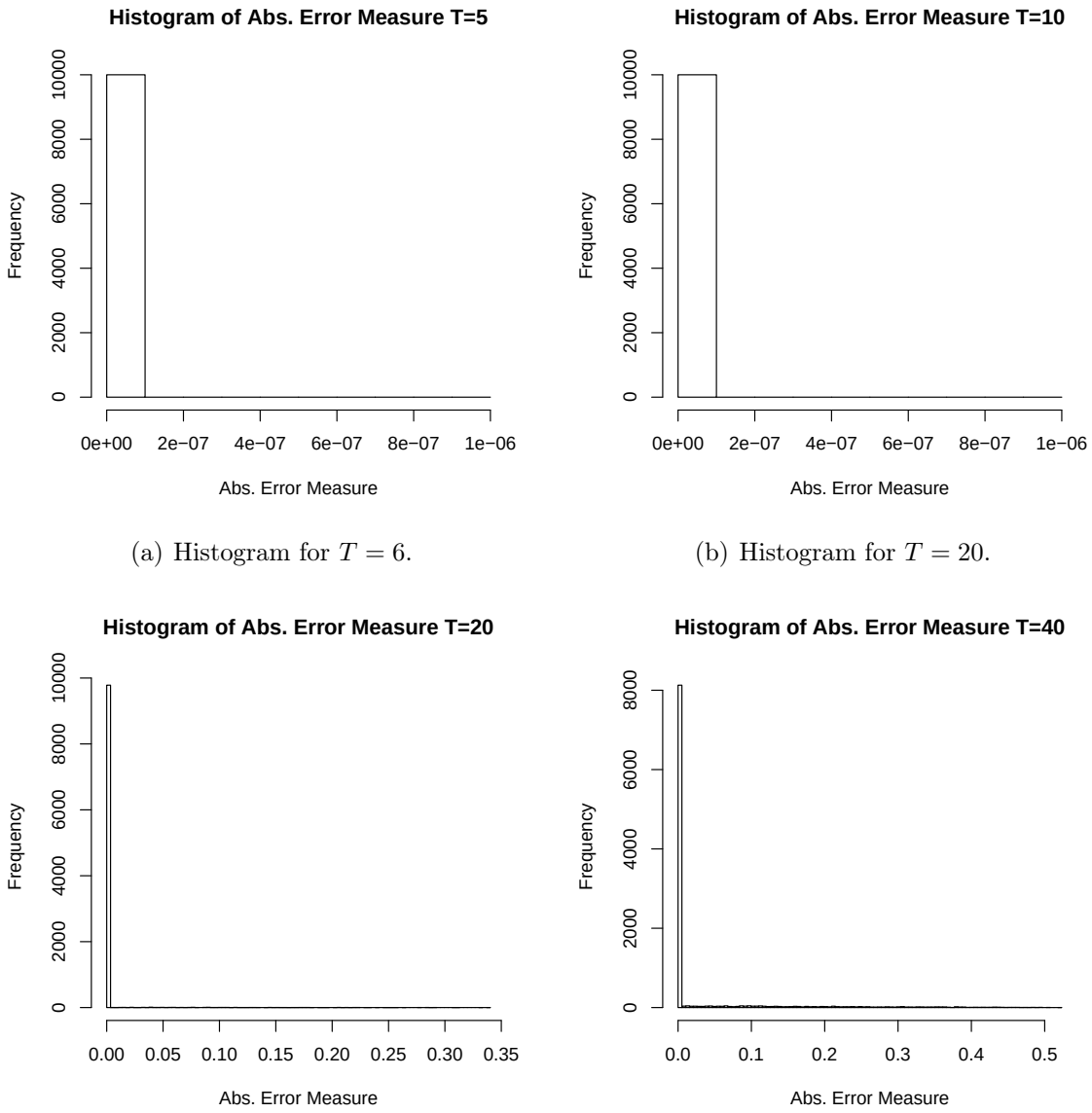


(a) Runtime per time stage t.

(b) Number of α-vectors over time stages t for the IP and α-min-Algorithm.

Figure 6.10.: Parameters per time step. We recognize that the runtime of the α-min-Algorithm increases with the number of α in $\tilde{\mathcal{V}}_t$.

longer for the problems with $T = 5, 10, 20, 40$. There are two possibly reasons for this behavior: First, the most time consuming part in the α-min-Algorithm is the back projection Algorithm which suffers from the high dimensionality of the observation space $\mathcal{O}$. Secondly, the high number of created α-vectors increases the runtime of the BL-Algorithm. As we see in Figure 6.10(a) the number of the α-vectors increases when we step back in the time. Therefore, the runtime of the back projection Algorithm increases, because we have to do many multiplication and addition operations with dense matrices. In opposite, the IP-Algorithm creates only a small number of α-vectors per time step. In our case this are two or less α-vectors as we see in Figure 6.10(b). This behavior increases the speed of the algorithm radically, because it keeps the size of the sets $\mathcal{V}_{t-1}^{a,o}$, $\mathcal{V}_t^a$ and $\mathcal{V}_t$ from equations (6.8) to (6.10) very small. We could get a runtime advantage for the α-min-Algorithm, if we are able to increase the efficiency of the back projection Algorithm 6.7 or decrease the maximal number N of α-vectors per time stage.



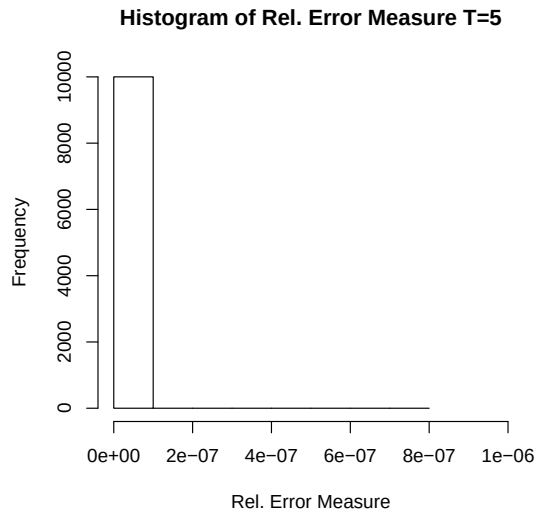
(a) Histogram for $T = 6$.

(b) Histogram for $T = 20$.

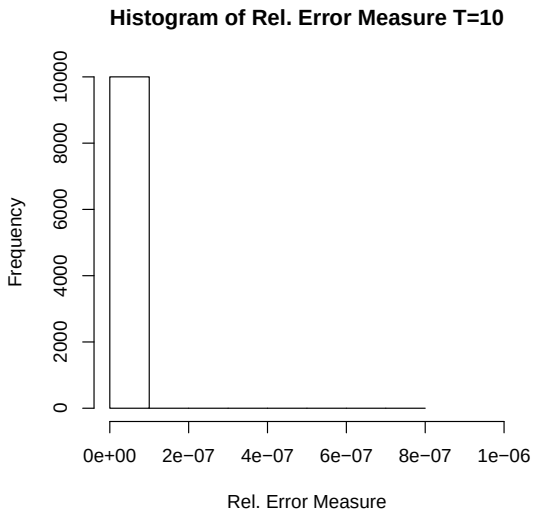
(c) Histogram for $T = 30$.

(d) Histogram for $T = 60$.

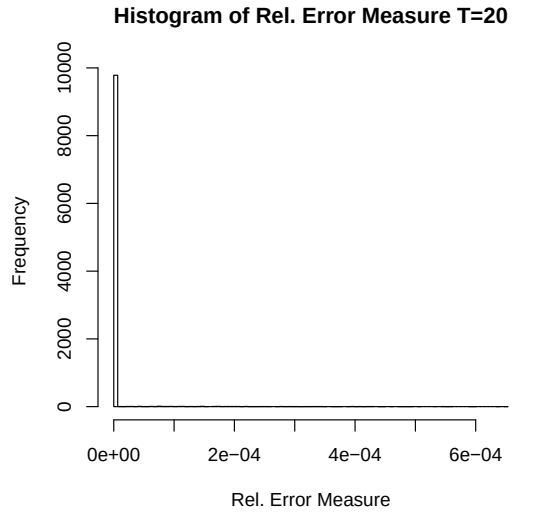
Figure 6.11.: Histograms according to the absolute error measure from equation (6.20). Every histogram presents the error measure for time step $t = 1$ and we chosen 10^4 belief states b randomly.



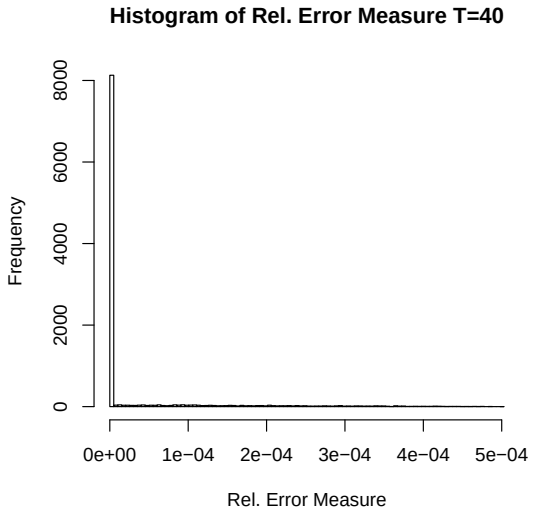
(a) Histogram for $T = 6$.



(b) Histogram for $T = 20$.



(c) Histogram for $T = 30$.



(d) Histogram for $T = 60$.

Figure 6.12.: Histograms according to the relative error measure from equation (6.21). Every histogram presents the error measure for time step $t = 1$ and we chosen 10^4 belief states b randomly.

7. Conclusion and Outlook

At the end of the this thesis, we draw conclusions from the presented work and we a give an outlook on future research work. The subjects presented in this work were different models to optimize maintenance scheduling of a gas turbine according to its life consumption. The models take into account that the gas turbine consists of different components which age at different rates. Further, the models capture the probabilistic nature of the different damage mechanisms which influence life consumption. At the end we get optimal maintenance strategies.

We present three different models which differ in the used service actions. We start with a simple replacement only model. In this case our service action sets the affected component always back into as new condition. Further, we only consider the risk of a total damage of the gas turbine. The modeling approach is similar to a risk life insurance. To solve this type of model, we first present a simple toy model. After this, we use an impulse control approach for a more complex and realistic model. This approach scales very well and is also suitable, if we increase the number of components.

As a next step, we extend the replacement model to an advanced replacement model. In this model the service action is still replacement, but we can repair the gas turbine after a failure. The repair action sets the effected component back into as new condition. We use a Dynamic Programing framework to solve this problem. We demonstrate that our model suffers from the curse of dimensionality. But we present an approximate Dynamic Programming approach to mitigate this behavior. In the ADP-Algorithm, we exploit the special structure of our problem and demonstrate for a two component model that we can reduce computational effort significantly with acceptable errors compared to the exact solution.

In the last step, we introduce the service action of inspection which is often used in real applications. In this case components are replaced prematurely, if they do not fulfill certain inspection criteria. In this case we gain information at every outage and we can readjust the failure probability. But the complexity increases, because we have no perfect state information like in the two models before. Therefore, we use a partially observable Markov Decision Process framework to model this type of service type. We develop a simple probabilistic model for crack growth under inspection to make use of the gain of information during an inspection. For the POMDP approach we also present an exact solution method, the IP-Algorithm, and an approximate method, the α-min-Algorithm, to solve the problem. Further, we show that the error of the approximate solution remains within tolerable bounds, but the runtime behavior of the approximate

algorithm is not so good as assumed.

An important result of the numerical case studies is that the optimal solution for the replacement and advanced replacement model is better than the standard maintenance schedule / policy which is used today. A Comparison for the inspection model was not carried out. In the replacement model we achieve a 56.30Mio. Euro higher revenue than in the standard maintenance case. The calculated optimal policies in the advanced replacement model are on average approximately 100.0Mio. Euro better than the standard policy.

The three mentioned models give us the possibilities to create an optimal maintenance schedule. In our case it is always a cost optimal maintenance model. It should be very easy to extend the model to different objective functions for example like availability or performance. This could be the next first step for future research on this topic. Also, we can use these models for every mechanical or electrical equipment which need maintenance.

Another interesting topic is to analyze how many life counters we really need. As mentioned in the thesis, an increase of the counters is equivalent to an increase of complexity. Therefore, it is important to choose the right number and location of life counter or to merge parts in groups together.

A main further research topic is the tuning and development of further approximate solution techniques for the approximate solution methods. In general we can explore the special structure of our problem in the Dynamic Programing framework to extend the model to more than two components. Further, we can explore the structure for different objective functions which should be quite the same to generate further approximate methods. In the POMDP framework respectively in the inspection model, one also should be able to tune the point based α-min-Algorithm, if one can find important belief states according to our maintenance pattern and the state transition function. Today, we can cover all three models in the POMDP framework, but it increases the complexity significantly. Therefore, a further important topic is a smart combination of the models introduced here.

As a last interesting future research topic, we mention the introduction of uncertainty in the operating regimes or in the financial data. In the presented work these two points are fixed by assumption, but in general they are scattered which can change the proposed outage schedule. This point also will increase the complexity of the models. In general the Dynamic Programming and POMDP approach are capable to handle such extensions as well.

A. Appendix - Stochastics

This subsection summarizes the necessary background information of stochastic theory and notation in an axiomatic way based on the work of [42]. First we deliver all necessary information about a random variable ω which describes the uncertainty in our models. We start with the definition of a σ-algebra which is state space of a random variable ω. It describes all possible results of a random experiment.

Definition A.1 (σ-algebra)
Let Ω be some set, and let $\mathcal{P}(\Omega)$ represent its power set. Then a subset $\mathcal{F} \subset \mathcal{P}(\Omega)$ is called a σ-algebra if it satisfies the following three properties:

- *$\mathcal{F}$ is non-empty: There is at least one $E \subset \Omega$ in $\mathcal{F}$.*
- *$\mathcal{F}$ is closed under complementation: If E is in $\mathcal{F}$, then so is its complement, $\Omega \setminus E$.*
- *$\mathcal{F}$ is closed under countable unions: If E_1, E_2, E_3, ... are in $\mathcal{F}$, then so is $E = E_1 \cup E_2 \cup E_3 \cup \ldots$.*

We call Ω the sample space and $\mathcal{F}$ is a family of events which are possible results of our random parameter ω. Next, we need two measure theoretical definitions to establish the term of probability space and distributions.

Definition A.2 (Measure & measurable space)
Let Ω be a set and $\mathcal{F}$ a σ-algebra over Ω. A function $\mu : \mathcal{F} \to \mathbb{R} \cup \{-\infty\} \cup \{+\infty\}$ is called a measure if it satisfies the following properties:

- *Non-negativity: For all $E \in \mathcal{F}$: $\mu(E) \geq 0$.*
- *Null empty set: $\mu(\emptyset) = 0$.*
- *σ-additivity: For all countable collections $\{E_i\}_{i \in \mathbf{N}}$ of pairwise disjoint sets in $\mathcal{F}$:*

$$\mu\left(\bigcup_{i \in \mathbf{N}} E_i\right) = \sum_{i \in \mathbf{N}} \mu(E_i).$$

A probability measure Pr, is a measure with total measure one $Pr(\otimes) = 1$. The pair $(\Omega, \mathcal{F})$ is called a measurable space, the members of $\mathcal{F}$ are called measurable sets. A triple $(\Omega, \mathcal{F}, \mu)$ is called a measure space.

The next theorem presents important properties of a probability measure Pr.

Theorem A.3 (Probability Measure)
For every probability measure Pr on a measure space $(\Omega, \mathcal{F}, \mu)$ we have the following properties for every event A, B, A_1, A_2, $\ldots \in \mathcal{F}$:

1. $Pr(\emptyset) = 0$,
2. *finite additivity:* $Pr(A \cup b) + Pr(A \cap B) = Pr(A) + Pr(B)$ *and it particularly* $Pr(A) + Pr(A^c) = 1$,
3. *monotony:* $A \subset B \Rightarrow Pr(A) \leq Pr(B)$,
4. *σ-subadditivity:* $Pr\left(\bigcup_{k\geq 1} A_k\right) \leq \sum_{k\geq 1} Pr(A_k)$,
5. *σ-continuity: If* $A_k \uparrow A$ *(e.g.* $A_1 \subset A_2 \subset \ldots$ *and* $A = \bigcup_{k=1} A_k$*) or* $A_k \downarrow A$*, then it follows* $Pr(A_k) \to Pr(A)$ *for* $k \to \infty$.

Next, we define a measurable function:

Definition A.4 (Measurable function)
Let $(\Omega, \mathcal{F})$ and $\left(\hat{\Omega}, \hat{\mathcal{F}}\right)$ be measurable spaces. A function $f\colon \mathcal{F} \to \hat{\mathcal{F}}$ is said to be measurable if the preimage under f is in $\mathcal{F}$ for every $\hat{E}$ in $\hat{\mathcal{F}}$, i.e.

$$f^{-1}\left(\hat{E}\right) := \left\{\omega \in \Omega\colon f(\omega) \in \hat{E}\right\} \in \mathcal{F} \quad \text{for all} \quad \hat{E} \in \hat{\mathcal{F}}.$$

If we put these pieces together, then we get a triple $W = (\Omega, \mathcal{F}, P)$ which is our probability space. In particular we obtain:

Definition A.5 (Probability Space)
A probability space W is a triple $W = (\Omega, \mathcal{F}, P)$:

- *Ω is its sample space*
- *$\mathcal{F} \subset \mathcal{P}(\Omega)$ is its σ-algebra of events*
- *P is its probability measure*

Now, we concentrate on the probability measure which describes a random variable. A random variable is defined by

Definition A.6 (Random variable)
A random variable $X\colon \Omega \to E$ is a measurable function from the set of possible outcomes Ω to some set $E \subset \mathbb{R}$. Ω belongs to a probability space $W = (\Omega, \mathcal{F}, P)$.

Conditional Probabilities

We present a brief introduction of conditional probabilities, confer [42]. In general we can say: A conditional probability measures the probability of an event given that another event has occurred. We give T. Bayes definition:

Definition A.7 (Conditional Probability)
Given two events A and B from the sigma-algebra of a probability space with $Pr(B) > 0$, the conditional probability of A given B is defined as the quotient of the probability of the join of events A and B, and the probability of B:

$$Pr(A|B) = \frac{Pr(A \cap B)}{Pr(B)}.$$

If A and B are two stochastic independent events, we get

$$Pr(A|B) = \frac{Pr(A \cap B)}{Pr(B)} = \frac{Pr(A)\,Pr(B)}{Pr(B)} = Pr(A)$$

and for the case of depending events Bayes formula says that

$$Pr(A|B) = \frac{Pr(B|A)\,Pr(A)}{Pr(B)}.$$

List of Abbreviation and Symbols

3d	3d dimensional.
ADP	Approximate Dynamic Programming.
CMB	Coffin-Manson-Basquin.
DP	Dynamic Programming.
EBH	Equivalent baseload hours.
EOH	Equivalent operating hours.
ES	Equivalent starts.
FBB	Find best belief (state).
HCF	High cycle fatigue.
KKT	Karush-Kuhn-Tucker-Condition.
LICQ	Linear independence constraint qualification.
LP	Linear program.
LCF	Low cycle fatigue.
MDP	Markov Decision Process.
MILP	Mixed integer linear program.
NLP	Non linear program.
ODE	Ordinary differential equations.
POMDP	Partially observable Markov Decision Process.
PWLC	Piecewise linear and convex.
PPP	Poisson Point Process.
PoD	Probability of detection.
PoE	Probability of premature exchange.
PoF	Probability of failure.

QP	Quadratic program.
λ, μ	Adjoint variables / Lagrange multiplier.
$\tilde{\mathcal{V}}_t$	Approximated set of α-vectors.
$\mathcal{V}_t$	Set of α-vectors.
$\Pi(\mathcal{S})$	Belief state space associated with $\mathcal{S}$.
$\partial\Omega$	Boundary of the set $\Omega \subset \mathbb{R}^n$.
$P(A\|B), p_{ij}$	Conditional probability.
$\delta_{\{\cdot\}}$	Delta distribution.
$f'(t)$	Time derivate of f.
i_{eff}	Discount factor.
$\mathbb{E}[\cdot]$	Expectation operator.
e, $\exp(\cdot)$	Exponential function.
$F(t), f(t), S(t)$	Failure distribution, failure density, survival distribution.
$K_{1\text{C}}$	Fracture toughness.
$\mathcal{H}$am	Hamilton function associated with a control problem.
$h(t)$, $H(t)$	Hazard rate, cumulative hazard rate.
$\mathcal{IH}$am	Impulse Hamilton function associated with a impulse control problem
$\chi_{\{\cdot\}}$	Indicator function.
∇f	Jacobian matrix / gradient of function f.
$\mathcal{L}$	Lagrange function.
$c(t)$	Life time counter.
$\hat{h}$	Mesh size.
$J(\cdot)$	Objective function.
$O(o,a,s), Pr(o\|s,a)$	Observation probability.
$x^\star$	Optimal solution or value marked by a $\star$.
$\frac{\partial}{\partial x_i} f(x)$	i-th partial derivative of f.
$\pi_t(x_t)$	Policy for time step t.
G_0 or G_T	Terminal reward function.
G_{I}	Jump reward function.
G / G_t	Continuous / per time step reward function .

$\mathcal{A}$	Set of all actions.
$\mathbb{Z}$	Set of integers numbers.
$\mathbb{N}$	Set of natural numbers.
$\mathcal{O}$	Set of observations.
$\mathbb{Q}$	Set of Rational numbers.
$\mathbb{R}^n$	Set of all real valued n-dimensional vectors.
$\mathbb{R}^{m\times n}$	Set of all real valued $m \times n$ matrices.
$\mathbb{R}_+$	All positive real numbers.
$\mathcal{S}$	Set of all states.
$\Gamma(t, x, \lambda)$	Shifting function.
$C^k(\Omega)$	Space of k times continuously differentiable functions on Ω.
Ω_u	Control space.
$L^p((a, b), \Omega)$	Lebesgue space on $\Omega \subset \mathbb{R}^n$.
$W^{k,p}((a, b), \Omega)$	Sobolev space on $\Omega \subset \mathbb{R}^n$.
Ω_x	State space.
$\mathbf{P}$, $\mathbf{P}(a)$	Stochastic matrix.
$\sigma(\cdot)$	Stress field.
K_{I}	Stress intensity factor.
T_{end}	Time horizon.
T	Number of time steps.
T_0	Time horizon beginning.
$p(s, a, s'), Pr(s'\|s, a), p_{ij}(a)$	Transition probabilitiy.
$x^{\mathrm{T}}, A^{\mathrm{T}}$	Transpose of vector x, matrix A.
$V_t(x_t)$	Value function for time step t.
m, η	Weibull parameter: Shape and scale.

Bibliography

[1] W. Alt. *Nichtlineare Optimierung: Eine Einführung in Theorie, Verfahren und Anwendungen*. Vieweg + Teubner, Wiesbaden, 2. edition, 2011.

[2] N. Andréasson. *Optimization of opportunistic replacement activities in deterministic and stochastic multi-component systems*. Chalmers University of Technology, 2004.

[3] G. B. Arfken. *Mathematical methods for physicists*. Elsevier, Acad. Press, Amsterdam, 6. edition, 2005.

[4] K. Arrow, T. Harris, and J. Marschak. Optimal inventory policy. *Econometrica (pre-1986), Jul 1951, Vol.19(3), p.250*, 19(3):250–null, 1951.

[5] Z. Artstein. Discrete and continuous bang-bang and facial spaces or: look for the extreme points. *SIAM Review*, 22(2):172–185, 1980.

[6] K. J. Åström. Optimal control of Markov Processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10:174–205, January 1965.

[7] S. B. Batdorf and J. G. Crose. A statistical theory for the fracture of brittle structures subjected to nonuniform polyaxial stresses. *Journal of Applied Mechanics*, 41(2):459–464, 1974.

[8] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1 edition, 1957.

[9] M. Berkelaar, K. Eikland, and P. Notebaert. lp_solve - open source (mixed-integer) linear programming system, 2004.

[10] D. P. Bertsekas. *Dynamic programming and optimal control. 1*. Athena Scientific, Belmont, Mass., 3. edition, 2005.

[11] D. P. Bertsekas. *Dynamic programming and optimal control. 2*. Athena Scientific, Belmont, Mass., 3. edition, 2007.

[12] A. Blaquiere. Differential games with piece-wise continuous trajectories. In P. Hagedorn, H. Knobloch, and G. Olsder, editors, *Differential Games and Ap-*

plications, volume 3 of *Lecture Notes in Control and Information Sciences*, pages 34–69. Springer Berlin Heidelberg, 1977.

[13] G. Bohrenkämper. Modernisierungen im gasturbinenservice. In *Stationäre Gasturbinen*, pages 1083–1099. Springer, 2010.

[14] M. Bolten, H. Gottschalk, and S. Schmitz. Minimal failure probability for ceramic design via shape control. *Journal of Optimization Theory and Applications*, 166(3):983–1001, 2015.

[15] F. Bookstein, D. Grass, J. Caulkins, G. Feichtinger, G. Tragler, and D. Behrens. *Optimal Control of Nonlinear Processes: With Applications in Drugs, Corruption and Terror*. JSTOR, 2011.

[16] D. Braess. *Finite Elemente : Theorie, schnelle Loeser und Anwendungen in der Elastizitaetstheorie*. Springer Berlin Heidelberg, Berlin, Heidelberg, 5. aufl. edition, 2013.

[17] R. Carnell. *lhs: Latin Hypercube Samples*, 2016. R package version 0.13.

[18] A. Cassandra, M. L. Littman, and N. L. Zhang. Incremental pruning: A simple, fast, exact method for partially observable markov decision processes. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pages 54–61. Morgan Kaufmann Publishers Inc., 1997.

[19] A. R. Cassandra. *Exact and Approximate Algorithms for Partially Observable Markov Decision Processes*. PhD thesis, Brown University, Providence, RI, USA, 1998. AAI9830418.

[20] A. Cayley. *The collected mathematical papers of Arthur Cayley. 9*. Univ. Press, Cambridge, 1. reprint. [der ausg.] cambridge 1896 edition, 1896.

[21] M. Chahim, R. Brekelmans, D. Den Hertog, and P. Kort. An impulse control approach to dike height optimization. *Optimization Methods and Software*, 28(3):458–477, 2013.

[22] M. Chahim and D. Grass. *Numerical Algorithms for Deterministic Impulse Control Models with Applications*. Econometrics, Netherlands, Europe, 2012.

[23] M. Chahim, R. Hartl, and P. M. Kort. *The Deterministic Impulse Control Maximum Principle in Operations Research: Necessary and Sufficient Optimality Conditions*. CentER Discussion Paper Series, 2011.

[24] M. Chahim, R. F. Hartl, and P. M. Kort. A tutorial on the deterministic impulse control maximum principle: Necessary and sufficient optimality conditions. *European Journal of Operational Research*, 219(1):18–26, 2012.

[25] C. Childs, J. Janawitz, and J. Masso. *Heavy-Duty Gas Turbine Operating and Maintenance Considerations*. GE Power & Water, Atlanta, 2015.

[26] P. G. Ciarlet. *Mathematical elasticity. 1. Three-dimensional elasticity*. North-Holland, Amsterdam, 1988.

[27] U. H. Clormann and T. Seeger. *Rainflow-HCM: ein Hysteresisschleifen-Zähl-algorithmus auf werkstoffmechanischer Grundlage*. Technische Hochschule Darmstadt, Fachgebiet Werkstoffmechanik, Darmstadt, 1985.

[28] P. Deuflhard and F. Bornemann. *Gewöhnliche Differentialgleichungen*. Walter de Gruyter, Berlin, 2013.

[29] J. S. Dibangoye, C. Amato, A. Doniec, and F. Charpillet. Producing efficient error-bounded solutions for transition independent decentralized MDPs. In *Proceedings of the Twelfth International Conference on Autonomous Agents and Multiagent Systems*, pages 539–546, May 2013.

[30] A. W. Drake. *Observation of a Markov process through a noisy channel*. PhD thesis, Massachusetts Institute of Technology, 1962.

[31] Y. Dujardin, T. Dietterich, and I. Chades. α-min: a compact approximate solver for finite-horizon pomdps - proofs. `https://sites.google.com/site/ijcaialphamin/home/Alpha_Min_Proofs.pdf`. Accessed: 2016-04-14.

[32] Y. Dujardin, T. Dietterich, and I. Chades. α-min: a compact approximate solver for finite-horizon pomdps. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 2582–2588. AAAI Press, 2015.

[33] E. Dynkin. Controlled random sequences. *Theory of Probability & Its Applications*, 10(1):1–14, 1965.

[34] L. A. Escobar and W. Q. Meeker. *Statistical methods for reliability data*. John Wiley & Sons, 2014.

[35] M. R. Eslami and R. B. Hetnarski. *Thermal Stresses - Advanced Theory and Applications*. Springer Netherlands, Dordrecht, 2009.

[36] European Energy Exchange AG. Historical data for EEX baseload hours prices. Database / Website, `http://www.eex.com/`, 2016.

[37] B. Fedelich. A stochastic theory for the problem of multiple surface crack coalescence. *International Journal of Fracture*, 91(1):23–45, 1998.

[38] D. Fett and D. Munz. *Mechanische Eigenschaften von Keramik*. Springer, Berlin, 1989.

[39] W. H. Fleming and R. W. Rishel. *Deterministic and stochastic optimal control*, volume 1. Springer Science & Business Media, 2012.

[40] O. Forster. *Analysis 1: Differential- und Integralrechnung einer Veränderlichen*. Vieweg+Teubner Verlag, Wiesbaden, 10., überarbeitete und erweiterte auflage edition, 1981.

[41] H. P. Geering. *Optimal control with engineering applications*. Springer, Berlin, 2007.

[42] H. Georgii. *Stochastik: Einführung in die Wahrscheinlichkeitstheorie und Statistik*. De-Gruyter-Lehrbuch. Walter de Gruyter, Berlin, 3. edition, 2007.

[43] M. Gerdts. *Optimal control of ODEs and DAEs*. Walter de Gruyter, 2012.

[44] G. Glockner. *Parallel and Distributed Optimization with Gurobi Optimizer*. Gurobi Optimization, Inc., Bad Homburg, Germany, 2015.

[45] H. Gottschalk and S. Schmitz. Optimal reliability in design for fatigue life. *SIAM Journal on Control and Optimization*, 52(5):2727–2752, 2014.

[46] Gurobi Optimization Inc. Gurobi optimizer reference manual, 2015.

[47] K. Habel, R. Grasman, R. B. Gramacy, A. Stahel, and D. C. Sterratt. *geometry: Mesh Generation and Surface Tesselation*, 2015. R package version 0.3-6.

[48] M. Hauskrecht. Value-function approximations for partially observable markov decision processes. *Journal of Artificial Intelligence Research*, pages 33–94, 2000.

[49] M. R. Hestenes. Variational theory and optimal control theory. In *Computing Methods in Optimizing Problems*, volume 1, page 1, 1964.

[50] R. Howard. *Dynamic Programming and Markov Processes*. Technology Press Research Monographs. MIT Press, 1960.

[51] A.-D. Ioffe, K. Makowski, and V.-M. Tihomirov. *Theory of extremal problems.*, volume 6. North-Holland Publishing Co., University of Tokyo Press, Amsterdam, 1978.

[52] R. Isaacs. *Differential games: A mathematical theory with applications to warfare and pursuit, control and optimization*. Krieger, Huntington, NY, reprint. with corr. and supplementary material edition, 1975.

[53] D. Jungnickel. *Optimierungsmethoden: Eine Einführung*. Springer, Berlin, Heidelberg, 3. edition, 2014.

[54] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in

partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134, 1998.

[55] A. Klenke. *Wahrscheinlichkeitstheorie*, volume 1. Springer, Berlin, Heidelberg, 2006.

[56] M. Koller. *Stochastische Modelle in der Lebensversicherung*. Springer-Verlag, Berlin, Heidelberg, 2010.

[57] C. Lechner and J. Seume. *Stationäre Gasturbinen*. Springer, Heidelberg, 2. edition, 2010.

[58] M. L. Littman. The witness algorithm: Solving partially observable markov decision processes. *Brown University, Providence, RI*, 1994.

[59] P. Massé. *Les Réserves et la régulation de l'avenir dans la vie économique: Avenir déterminé*. Number Bd. 1 in Actualités scientifiques et industrielles. Hermann, 1946.

[60] M. D. McKay. Latin hypercube sampling as a tool in uncertainty analysis of computer models. In *Proceedings of the 24th Conference on Winter Simulation*, WSC '92, pages 557–564, New York, NY, USA, 1992. ACM.

[61] L. Milović, T. Vuherer, Z. Radaković, B. Petrovski, M. Janković, M. Zrilić, and D. Daničić. Determination of fatigue crack growth parameters in welded joint of hsla steel. *Structural integrity and life, UDK*, 620(24):669–15, 2011.

[62] M. A. Miner et al. Cumulative damage in fatigue. *Journal of applied mechanics*, 12(3):159–164, 1945.

[63] G. E. Monahan. State of the art survey of partially observable markov decision processes: theory, models, and algorithms. *Management Science*, 28(1):1–16, 1982.

[64] A. Myers. *Complex System Reliability: Multichannel Systems with Imperfect Fault Coverage*. Springer-Verlag London, London, 2nd edition edition, 2010.

[65] T. Nakagawa. *Maintenance theory of reliability*. Springer Science & Business Media, London, 2005.

[66] J. Nascimento and W. Powell. An optimal approximate dynamic programming algorithm for the economic dispatch problem with grid-level storage. *IEEE Transactions on Automatic Control (to appear in print)*, 2013.

[67] A. Palmgren. Die lebensdauer von kugellagern. *Zeitschrift des Vereins Deutscher Ingenieure*, 68(14):339–341, 1924.

[68] P. Paris and F. Erdogan. A critical analysis of crack propagation laws. *Journal of basic engineering*, 85(4):528–533, 1963.

[69] G. A. Pavliotis. Stochastic processes and applications. *Diffusion Processes, the Fokker-Planck*, 2014.

[70] R. Perez and D. Lawhon. *Operator s Guide to General Purpose Steam Turbines: An Overview of Operating Principles, Construction, Best Practices, and Troubleshooting*. Wiley, 2016.

[71] H. J. Pesch. *Schlüsseltechnologie Mathematik: Einblicke in aktuelle Anwendungen der Mathematik*. Springer-Verlag, 2002.

[72] L. S. Pontryagin. Optimal control processes. *Usp. Mat. Nauk*, 14(3), 1959.

[73] W. B. Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*, volume 703. John Wiley & Sons, Hoboken, NJ, 2007.

[74] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. Wiley, Hoboken, NJ, 2005.

[75] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.

[76] D. Radaj. *Ermüdungsfestigkeit: Grundlagen für Leichtbau, Maschinen-und Stahlbau*. Springer-Verlag, 2013.

[77] R. Rempala and J. Zabczyk. On the maximum principle for deterministic impulse control problems. *Journal of optimization theory and applications*, 59(2):281–288, 1988.

[78] H. A. Richard and M. Sander. *Ermüdungsrisse*. Vieweg+Teubner Verlag / GWV Fachverlage GmbH, Wiesbaden, Wiesbaden, 2009.

[79] R. G. V. Richard F. Hartl, Suresh P. Sethi. A survey of the maximum principles for optimal control problems with state constraints. *SIAM Review*, 37(2):181–218, 1995.

[80] S. Schmitz. *A Local and Probabilistic Model for Low-Cycle Fatigue: New Aspects of Structural Analysis*. Series in Computational Science. Hartung-Gorre, 2014.

[81] S. Schmitz, H. Gottschalk, G. Rollmann, and R. Krause. Risk estimation for lcf crack initiation. In *ASME Turbo Expo 2013: Turbine Technical Conference and Exposition*, pages V07AT27A007–V07AT27A007. American Society of Mechanical Engineers, 2013.

[82] S. Schmitz, T. Seibel, T. Beck, G. Rollmann, R. Krause, and H. Gottschalk. A probabilistic model for lcf. *Computational Materials Science*, 79:584–590, 2013.

[83] R. Schöbi and E. Chatzi. Maintenance planning under uncertainties using a

continuous-state pomdp framework. In *Model Validation and Uncertainty Quantification, Volume 3*, pages 135–143. Springer, 2014.

[84] A. Shiryaev. On the theory of decision functions and control of a process of observation based on incomplete information. *Selected Translations in Math. Stat. and Prob*, 6:162–188, 1966.

[85] Siemens AG. Flexible performance, convincing quality economical and future proof packages. Brochure, 2014.

[86] R. D. Smallwood and E. J. Sondik. The optimal control of partially observable markov processes over a finite horizon. *Operations Research*, 21(5):1071–1088, 1973.

[87] E. J. Sondik. The optimal control of partially observable markov processes. Technical report, DTIC Document, 1971.

[88] S. Suresh. *Fatigue of materials*. Cambridge Univ. Press, Cambridge, 2. edition, 2006.

[89] L. Tierney, A. J. Rossini, N. Li, and H. Sevcikova. *snow: Simple Network of Workstations*, 2013. R package version 0.3-13.

[90] A. C. Ugural. *Advanced strength and applied elasticity*. Prentice Hall, Upper Saddle River, NJ, 4. edition, 2003.

[91] R. Vinter. *Optimal control*. Springer Science & Business Media, Boston, 2010.

[92] A. Waechter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[93] A. Wald. *Sequential analysis*. Dover Publ., New York, unabr. and unaltered republ. [d. ausg.] new york 1947 edition, 1947.

[94] W. Weibull. *A statistical theory of the strength of materials*. Generalstabens litografiska anstalts förlag, 1939.

[95] W. Weibull. A statistical distribution function of wide applicability. *Journal of applied mechanics*, 103:33, 1951.

[96] J. Werner. *Numerische Mathematik: Band 1: Lineare und nichtlineare Gleichungssysteme, Interpolation, numerische Integration*, volume 32. Vieweg, Braunschweig, 2013.

[97] A. Wöhler. Bericht über die versuche, welche auf der königl. niederschlesisch-märkischen eisenbahn mit apparaten zum messen der biegung und verdrehung

von eisenbahnwagen-achsen während der fahrt angestellt wurden. *Zeitschrift für Bauwesen*, 8(1858):641–652, 1858.

[98] N. L. Zhang and W. Liu. Planning in stochastic domains: Problem characteristics and approximation. Technical report, Technical Report HKUST-CS96-31, Hong Kong University of Science and Technology, 1996.

[99] W. Zulehner. *Numerische Mathematik: Eine Einführung anhand von Differentialgleichungsproblemen Band 2: Instationäre Probleme*. Birkhäuser, Basel, 2011.

Declaration of Authorship

I hereby declare that the thesis submitted is my own unaided work. All direct or indirect sources used are acknowledged as references.

This thesis was not previously presented to another examination board and has not been published.

Michael Gröger

Mülheim, March 5, 2017