After-Sales Services During an Asset’s Lifetime: Collaborative Planning of System Upgrades

Fiona Sloothaak, Alp Akçay, Geert-Jan van Houtum, Matthieu van der Heijden

To cite this article:
Fiona Sloothaak, Alp Akçay, Geert-Jan van Houtum, Matthieu van der Heijden (2023) After-Sales Services During an Asset’s Lifetime: Collaborative Planning of System Upgrades. Service Science

Published online in Articles in Advance 14 Mar 2023

https://doi.org/10.1287/serv.2023.0318

Full terms and conditions of use: https://pubsonline.informs.org/Publications/Librarians-Portal/PubsOnLine-Terms-and-Conditions

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article’s accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2023, INFORMS

Please scroll down for article—it is on subsequent pages
After-Sales Services During an Asset’s Lifetime: Collaborative Planning of System Upgrades

Fiona Sloothaak, a Alp Akçay, a Geert-Jan van Houtum, a, b Matthieu van der Heijden b

a Eindhoven University of Technology, 5600 MB Eindhoven, Netherlands; b University of Twente, 7500 AE Enschede, Netherlands

Corresponding author

Contact: f.sloothaak@tue.nl, https://orcid.org/0000-0002-1722-7070 (FS); a.e.akcay@tue.nl, https://orcid.org/0000-0003-4225-5434 (G-JvH); m.c.vanderheijden@utwente.nl,

Received: September 29, 2022
Revised: December 14, 2022
Accepted: December 31, 2022
Published Online in Articles in Advance: March 14, 2023

https://doi.org/10.1287/serv.2023.0318

Abstract. We consider a physical asset consisting of complex systems, where the systems may require upgrades during the lifetime of the asset. In practice, the asset owner and system supplier can make the upgrade decisions together, requiring a decision-support model that can be jointly used to optimize the total benefit for both parties. Motivated by a real-life use case including an asset owner and a system supplier, we build a continuous-time model to optimize the upgrade decisions of a system during the fixed lifetime of the asset. In our model, we capture the key critical factors that drive the upgrade decisions: increasing functionality requirements due to evolving technology, age-dependent maintenance costs, a predetermined overhaul plan of the asset, and the lifetime of the asset. A system upgrade is less costly if it is executed jointly with an asset overhaul. We first analyze the case with no additional cost of upgrading outside an overhaul. We analytically characterize the structure of the optimal upgrade policy under various realistic assumptions that lead to different types of cost functions. We then use these results as a building block to characterize the optimal policy for a generalized cost function. When there is a penalty for upgrading outside an overhaul moment, we propose a dynamic programming approach that efficiently determines the optimal upgrade policy by using our analytical results. We also prove that as this penalty increases, the optimal policy can only change to one where the number of upgrades not jointly executed with overhauls is reduced. However, the optimal number of upgrades is a nonincreasing function of this penalty. Also, surprisingly, more overhauls may lead to a smaller number of upgrades under the optimal policy.

Funding: This publication is part of the project “Maritime Remote Control Tower for Service Logistics Innovation (MARCONI)” (project 439.18.309) of the research program “Integrator-Logistics as Enabler for Enhancing Society,” which is (partly) financed by the Dutch Research Council (NWO).

Keywords: maintenance • upgrade policy • technology advancement • age-dependent cost functions • obsolescence

1. Introduction

Maintaining a satisfactory operational level for physical assets—that is, capital goods—during their entire lifetime is a very challenging problem (Pierskalla and Voelker 1976, Arts et al. 2019). Assets typically have complex designs with many different systems installed. Each system itself often consists of many components that require replacement in case of failures. The more complex and specialized the systems, the more common it is to see a closer engagement between system suppliers and asset owners, typically in the form of a service agreement that delegates certain roles to the system supplier (e.g., procurement of replacement components or maintenance of the system), referred to as after-sales services (Li and Tomlin 2022). Because assets are typically used for a long time, the systems in it can become obsolete and require upgrades. The decision on when to upgrade a system is a complex one, with inputs necessary from both the asset owner and the system supplier.

Especially for complex systems customized and serviced by a system supplier in the form of a partnership with an asset owner, there is an interest to make the system-upgrade decisions in a collaborative way, and this requires a decision-support model to plan the system upgrades as part of after-sales services.

Our work is motivated by a real-life problem as part of a collaboration with an owner of moving assets in a maritime setting (e.g., a frigate, or a type of small and fast military ship) and an original equipment manufacturer who designs, builds, and services a complex system in these assets (e.g., a radar system installed on a frigate). Frigates are typically in use for over 30 years, whereas the incorporated systems themselves do not necessarily last several decades. To ensure that a frigate works properly, a schedule is created upfront with designated moments (i.e., overhauls) where the frigate is docked and large-scale maintenance is executed. Docking a ship is costly, and it can therefore be efficient to
perform the upgrade of a system at such an overhaul moment. For instance, by performing an upgrade of a radar system together with a frigate overhaul, the “setup” required to upgrade the radar system can be saved. On the other hand, the costs for sustaining the current system may increase over time, leading to the question of when to perform system upgrades to strike the trade-off between the costs of upgrades and the costs associated with using an existing system in the presence of a predetermined asset-overhaul schedule.

Based on extensive discussions with our industry partners, we came to the three dominant factors that drive the costs for sustaining a system: functionality requirements, the age-dependent behavior of failure frequency and maintenance costs, and the asset’s lifetime. We elaborate on these factors in more detail next. First, advancement of technologies leads to systems that perform much better than the older ones, which may cause higher expectations of a system’s functionality. That is, the asset owner experiences a disutility of not using a system with the most recent technology. Examples include the desire for a more energy-sustainable system, the wish to rival or even surpass the functionality that competitors have, and more. This aspect is studied in some papers (e.g., Nair and Hopp 1992 and Hopp and Nair 1994), but is surprisingly often neglected in existing literature on maintenance strategies (de Jonge and Scarf 2020).

Secondly, the components of a system may fail over time, requiring corrective maintenance so that the system can continue its operation as intended. We note that older systems may require more corrective maintenance, known as age-dependent failure behavior (Wang 2002, Zhao et al. 2017). Also, all maintenance activities occur with associated costs, which are stationary in classical variants of age-based maintenance models (Barlow and Hunter 1960). Naturally, age-dependent costs are often necessary to model reality more closely. Several factors contribute to the need for age-dependent costs, such as salvage value and failure costs. Regarding the salvage value, we observe that the residual value of the system in use typically decreases as newer and improved versions come to market. The failure costs can also be age-dependent, especially when assets have relatively long lifetimes. This is often due to obsolescence issues—that is, spare components become costlier or even unavailable over time because newer versions are developed and the production of older components or systems is discontinued. Moreover, trained technicians with the knowledge and skills to execute upgrades may no longer be available. A well-chosen strategy can reduce the risk of (unexpected) obsolescence issues and their corresponding costs (International Electrotechnical Commission 2019), such as monitoring for obsolescence, last-time-buys (Behfard et al. 2015), or partner agreements—for example, incorporation of availability guarantees from suppliers for critical components (Tomczykowski 2003). Nevertheless, the level of uncertainty and associated costs make the adoption of these strategies difficult and can lead to using system upgrades as a means to tackle the increasing sustainment costs.

Finally, the timing of an upgrade may be affected by the remaining lifetime of the asset. Existing literature often focuses on long-run expected average costs for systems with an infinite time horizon. However, an infinite horizon rarely applies to real-life settings: As the end of the asset’s lifetime is near, it makes little sense to execute large-scale upgrades or improvements anymore. Moreover, the total costs during the lifetime can be reduced by bringing forward (or postponing) certain maintenance activities.

In this paper, we consider a continuous-time model that aims to find an optimal upgrade policy for a single system in an asset with a predetermined overhaul planning. The main novelty of our work is to bring all three factors mentioned above in one model in order to address the planning problem of system upgrades as realistically as possible. Also, to the best of our knowledge, our model is the first in the age-based maintenance literature with a schedule of low-cost maintenance moments (i.e., overhaul moments for system upgrades) and age-dependent operating costs (i.e., the age-dependent penalty of not using the system with the most recent technology). For this model, we analytically characterize the structure of the optimal upgrade policy by first considering a special case with no additional penalty of upgrading outside an overhaul moment. We do this first for some special forms of the cost functions, and then, by using these results as a building block, we characterize the structure of the optimal upgrade policy under a generalized cost structure. This allows for a relatively simple characterization of the optimal upgrade policies and can be used to design an efficient solution approach based on dynamic programming for the case where the penalization for upgrading outside an overhaul moment is also included. Finally, we establish sensitivity results with respect to the input parameters. For example, it is intuitively clear that the optimal number of upgrades does not increase as the price for the upgrades increases. However, surprisingly, this is not necessarily true with respect to many other input parameters, such as the penalty for upgrading outside overhauls, the number of overhauls, or a more rapid increase of failure costs.

The remainder of this paper is organized as follows. In Section 2, we review related literature, and in Section 3, we introduce our model formally. In Section 4, we derive high-level insights for the structure of the optimal policy if upgrade executions outside overhauls are not penalized. For the general case, we use the results in Section 4 to introduce a solution approach on how to calculate the optimal policy efficiently in Section 5. Finally, we derive several managerial insights in Section 6 by testing the dependency of the optimal policy with respect to the input parameters. We summarize our key findings and insights in Section 7.
2. Literature Review

Our work mainly relates to two streams of literature: (i) age-based maintenance models with age-dependent costs, and (ii) upgrade/replacement decision making by considering the effects of evolving technology. In this section, we review the related literature for both streams and describe the contributions of our paper.

The first stream concerns age-based maintenance models. In the classical variants of these models, we consider a single system with a stochastic lifetime distribution. The policy maker can choose to execute preventive maintenance against fixed costs that are lower than the costs that are incurred if the maintenance activity would be executed upon failure (Barlow and Hunter 1960). However, it is often the case that not all times are convenient to do maintenance on a system. Examples include systems in production lines, the aerospace or maritime industry, or other systems in industries where unplanned downs severely interrupt operations (Arts et al. 2019). In that case, the policy maker may choose to apply minimal repair until the planned maintenance moment is reached, meaning that the system is restored to a state right before the failure anytime a failure occurs.

There is a vast literature on these types of maintenance models (Wang 2002). We point out that in these papers, executing maintenance is often considered to be a replacement of the system. Initially, stationary costs are considered (Barlow and Hunter 1960, Barlow and Proschan 1965), but quickly, also age-dependent costs were accounted for in the models, as shown in Table 1. In Boland (1982) and Boland and Proschan (1982), a setting is described where the minimal repair costs increase with the number of occurrences of failures for both a finite and an infinite time horizon. A more general minimal repair-cost function is considered in Tilquin and Cléroux (1975), but replacement costs are assumed to be stationary. Nondecreasing minimal repair and replacement costs are considered in Segawa et al. (1992) for an infinite time horizon. Under age-dependent salvage values, Chien (2010) shows the existence and uniqueness of the optimal age for preventive replacement. A recent paper (Sanoubar et al. 2021) shows the existence of an optimal replacement policy for an infinite time horizon that allows for a wide class of age-dependent replacement-cost functions without minimal repair. Schouten et al. (2022) extend some classical age-based replacement policies with costs that vary in a cyclical manner. Literature that considers a finite time horizon is scarcer; Dagpunar and Jack (1994) determine the optimal number of imperfect preventive maintenance actions for a finite horizon, given that the minimal repair is made at any failure (with stationary costs).

This paper considers a continuous-time setting with nondecreasing upgrade costs, failure rate, and minimal repair costs, while accounting for the finite lifetime of the asset. It is important to highlight that the other papers mentioned in Table 1 build maintenance-optimization models at component level, whereas we do it at system level (i.e., we take replacement decisions for the system, but at the same time, consider a cost function including the maintenance of components within that system). A major novel aspect is that not only maintenance on system level is considered, but there already exists an

Table 1. Literature on Age-Based Maintenance Models with Age-Dependent Costs

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning horizon</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Finite (deterministic)</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Infinite</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Continuous</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Discrete</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Preventive action</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Perfect</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Imperfect</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Effect of corrective action</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>New condition</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Condition just before failure</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>(i.e., minimal repair)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Age-dependent costs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimal repair</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Corrective replacement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Preventive replacement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operating cost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schedule of low-cost preventive action</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
overhaul plan (marked as the schedule of low-cost preventive action moments in Table 1) upfront for the entire asset. This means that the execution of upgrades during the overhauls is typically preferred in order to prevent disruption of asset availability. Upgrades can still be planned at moments when there is no overhaul, but this comes with a penalty that portrays the severity of the disruption. The effects of such overhaul plans are unexplored in existing literature on age-based maintenance models. Furthermore, we are the first to consider age-dependent operating costs (to model the disutility of not using the most recent system) within this stream.

The second stream of literature relates to the effect of executing upgrades to meet (increasing) functional requirements. This aspect is studied already in a series of papers (Nair and Hopp 1992, Hopp and Nair 1994, Nair 1995), where a discrete-time model is proposed, in which one needs to decide whether to keep current technology or to move to another version at every time step. Through a Markov Decision Problem formulation, an optimal solution can be determined through dynamic programming, and also the infinite-horizon case can be solved through an efficient algorithm. Technology advancement is also considered in Rajagopalan et al. (1998), but for an entirely different environment, where technologies provide a certain capacity that is needed for operations. In this stream, it is common to assume an underlying stochastic model to represent the evolution of technology or the prediction of technological advancements. This is not necessary in our work, where we directly model the disutility of not using the system with the most recent technology. A continuous-time multicomponent maintenance model is considered in Mercier (2008), where the goal is to identify when to upgrade all components after a single technology improvement among the class of reactive upgrade policies. Analytical results are derived for a similar problem in Öner et al. (2015), but then from the perspective of a maintenance service provider who has to choose between preventive and corrective upgrading for a fleet of systems/assets after a redesign of a component. Nguyen et al. (2013) bridge technology change with spare-part inventory management in a discrete-time model and use dynamic programming to determine the optimal maintenance policy for a fixed horizon. Although this paper is rich in the different aspects that come with upgrades (i.e., potential loss of spare parts, different maintenance options, and so forth), no structural insights on the optimal policy are provided.

In this paper, we consider a continuous-time model similar to Mercier (2008). We restrict ourselves to an upgrade policy for a single system, but we allow for continuously evolving technology and preventive upgrade policies (with a minimal repair strategy to deal with failures). Our underlying costs and penalty functions are age-based, and we derive structural insights for the optimal upgrade policy. In addition, we incorporate overhaul moments where large-scale maintenance is executed, and, hence, it can be cost-effective to execute a system upgrade at such a moment.

3. Model Description

We consider an asset that is in operation for a finite (known) lifetime $H$. Typically, an asset has many different systems, subsystems, and components in place on different indenture levels. We take a simplified view, as illustrated in Figure 1. That is, an asset is composed of different systems, and, in turn, every system has many different (critical) components that are needed for an operational system. In practice, the upgrade decision of each system can be decoupled from each other. For example, a frigate as an asset has various systems (e.g., propulsion, navigation, radar, and weapon systems) that work independently. That is, the failures of a navigation system do not depend on the failures of a radar system. Also, each system is designed and serviced by a different supplier. Thus, the penalty of not using the most recent version of these systems can be modeled separately in collaboration with different suppliers. Therefore, we aim to determine a cost-optimal upgrade planning for a single system. We assume that all costs are expressed in monetary units at the start of the planning horizon (so we do not compute net present values). We provide a formal description of our model and notation next. We summarize our notation in Appendix A as well.

The upgrade of a system may interfere with the operations of the entire asset. It is therefore efficient to perform the upgrade of a system together with other asset-level, large-scale maintenance activities, referred to as overhauls. For example, when a frigate is already out-of-service for overhaul, any intervention to the frigate can be performed at no additional cost of taking the frigate out of service (which would be necessary during an upgrade). We assume that there exists an overhaul plan with planned moments for the entire asset. Overhaul periods are relatively short with respect to the total lifetime of the asset. Therefore, we allow for a simplification of the model by assuming that the overhaul takes no time in the planning, and, hence, the state of the system also has not changed after an overhaul, unless an

Figure 1. Asset Decomposition
upgrade had been planned. This yields a continuous-time model with given overhaul moments.

We denote the time between consecutive overhauls by $M_1, \ldots, M_m, M_{m+1}$, where $m$ is the number of overhauls during the asset’s lifetime $H$. In particular, $M_1$ denotes the time of the first overhaul, and $M_{m+1}$ denotes the duration between the last overhaul and the end of the asset’s lifetime. We allow for system upgrades to be executed at times when there is no overhaul, but we penalize these events with fixed costs $c_d$. Note that if a system upgrade can be executed easily and has no asset-level effect, the setting simplifies to $c_d = 0$. We refer to this special setting as the base case.

As mentioned in Introduction, we model several factors that drive an asset owner to upgrade a system through generalized costs and penalty functions. We explain this in more detail next. First, we say that a system upgrade comes with certain upgrade costs, for which we assume the following.

**Assumption 1.** The upgrade costs can be written as $c_0 - v(t)$, where $c_0 > 0$ can be seen as the price for purchasing and installing the upgraded version of the system, $v(t)$ corresponds to the salvage value of the current system, and $t \geq 0$ is the time since the current system is in use. Moreover, we assume that $c_0 > v(0)$ and that the salvage value $v(t)$ is nonincreasing and differentiable in $t$.

We point out that the price for a system upgrade $c_0$ is constant over time—that is, has no dependency on time. This independence reflects that asset owners are willing to pay a certain fixed price for the newest generation of a system, which is common for capital goods (Chien 2010). The asset owner may not be willing to pay a higher price than what was paid for an older system with the same asset functionality as when it was first used. From the seller’s (e.g., the system supplier who designs, produces, and services the system) point of view, increasing the price of a system along the way may lead to losing the account with the customer or be infeasible due to contractual agreements. The salvage value $v(t)$ corresponds to the possible costs or returns that come with the disposing of the current system in use and depends only on the time it has been in use. In particular, we make no assumptions on whether the salvage value is positive or negative. We exclude the unrealistic scenarios where $c_0 \leq v(0)$; otherwise, it would be profitable to continuously upgrade. The property that the salvage value is nonincreasing describes the natural phenomenon that the disposal of an older system becomes less profitable (or costlier) over time. Also, in cases where equipment is refurbished or reused, the salvage value can decrease in age because the refurbishment process may involve adjustments and additions of extra components (Sanoubar et al. 2021).

As soon as a certain system is installed in the asset, improved versions and upgrades may come to the market. Increasing functional requirements of the current system may lead to the necessity for upgrading to newer versions to meet the desired requirements. We model this effect through a penalty function $c_f(t)$, where $t$ denotes the time that the system is in use since its last upgrade. It represents the gap in functionality between the system in use and the upgraded version; see, for example, Sols et al. (2012) for an example of such a gap function. Note that, by definition, there is no gap when the system in use is brand-new—that is, $c_f(0) = 0$.

**Assumption 2.** The functionality gap function $c_f(t)$ is non-decreasing in $t$ with $c_f(0) = 0$, where $t$ denotes the time since the current system is in use—that is, the last upgrade.

In practice, the penalty function $c_f(\cdot)$ can be directly elicited from the system supplier (who may assess the condition of the latest technology and current capabilities) and the asset owner (who may assess how critical it is for the entire asset to exploit the benefits of the latest technology in a particular system). For example, the disutility of not using the system with the most recent technology can be a steeply increasing function of time if the system is safety-critical (e.g., a software module protecting the asset from failures, but becoming prone to cyberattacks over time), or it can be a constant or slowly increasing function if it performs the defined function adequately, regardless of its age (e.g., a furnace used for the heat-treatment of steel products). In some cases, historical data can be used to derive $c_f(\cdot)$. For instance, consider a piece of machining equipment, which requires a certain amount of power to perform its task when it is new. The wear caused by its usage may lead to an increase in the cutting force required in the process and, hence, the required cutting power. So, there is an influence of how long the machine has been used on energy consumption. Historical data can be used to identify a function that represents how the power required by the process changes with respect to the usage hours. Such a function can be translated into a penalty function $c_f(\cdot)$ that quantifies the additional energy cost as a function of time.

The system itself consists of many different (critical) components that are subject to failures. We aggregate the failures to a single failure rate for the system, where every failure comes with a certain expected cost. We consider the setting where failures do not occur less likely as time progresses, as well as that the associated costs do not lessen over time. Moreover, we assume that the repair or replacement of a component restores the total failure rate back to the value right before the failure event, also known as a minimal repair strategy (Barlow and Proschan 1965).

**Assumption 3.** Let $h(t)$ denote the failure rate of the system and $k(t)$ the expected repair costs to bring the system back to the state just prior to the failure, where $t \geq 0$ denotes
the time since the current system is in use. The functions \( h(t) \) and \( k(t) \) are both nondecreasing in \( t \). Repairs are immediate.

We point out that minimal repair is assumed due to the notion that the system is composed of many components, and the failure of one critical component will not (likely) affect the total failure rate of the system, as also argued in Asadi et al. (2023). For a particular system that is always sold at the same price, it can be argued that the quality level (i.e., reliability) of the systems will be similar across generations; hence, we assume the failure-rate function \( h(t) \) remains the same for each system. There is an extensive literature in reliability theory to estimate failure rate functions by using historical data (see, e.g., chapter 3 of Gertsbakh 2000). We note that the failure-rate function is best determined with the collaboration of the asset owner (who knows how intensively the system is used) and the system supplier (who has the expertise of system designers and/or can pool data from many different users of their systems). The associated expected costs that come with a failure have a general form, as long as it is nondecreasing in time. These costs are an aggregation of the actual repair and replacement costs and downtime costs, and particularly obsolescence issues for critical components. That is, the nondecreasing property of \( k(t) \) mostly stems from obsolescence issues that can arise when critical components are no longer in stock or produced, and the repair becomes costlier as a more expensive fix needs to be applied.

**Remark 1.** For a system with many critical different components, it may hold that the failure of the system relates to the failure of one particular component by a certain probability. In this example, the expected repair costs \( k(t) \) may be written as

\[
k(t) = \sum_{i=1}^{I} k_i(t)p_i,
\]

where \( I \) is the number of (critical) components in the system, \( p_i \) is the probability that the failure is due to the breakdown of component \( i \), and \( k_i(t) \) is the expected repair cost of a failed component \( i \). Note that the values of \( p_i \) can be determined by using historical data of component failures, and \( k_i(t) \) by using the costs associated with component procurement and repairs at different ages of the system, which can be obtained by, for example, monitoring the spare part prices on the market.

The goal is to find the policy that minimizes the total costs, which is the aggregation of all costs and penalties over the entire asset’s lifetime. To formalize this, we introduce the following notation. Under a given policy \( \Pi \), let \( N \) denote the number of upgrades, \( T_1 \) the first upgrade moment (if \( n > 0 \)), and \( T_2, \ldots, T_N \) the time between upgrade moments (if \( N \geq 2 \)). Write \( T_{N+1} = H - \sum_{i=1}^{N} T_i \) as the time between the last system upgrade (if any) and the end of the asset’s lifetime, which we also refer to as the remaining lifetime. Let a cycle be the time a single system version is in use—that is, the time between the moment the system is installed until it is disposed of, which is either at the next upgrade moment or at the end of the asset’s lifetime. We define the cycle costs \( C(T) \) as the aggregation of the salvage value, missing functionality penalty, and the expected costs that come with system failures in a cycle of length \( T \)—that is,

\[
C(T) = -v(T) + \int_0^T c_f(t) dt + \int_0^T k(t)h(t) dt.
\]

We point out that the price \( c_0 \) is not included in the cycle costs, as the last cycle covers a disposal, but not an upgrade. Under policy \( \Pi \), let \( S^{\Pi} \) denote the number of times a system upgrade is executed, but not jointly with an overhaul—that is,

\[
S^{\Pi} = \left\{ n \leq N : \sum_{i=1}^{n} T_i \neq \sum_{i=1}^{j} M_i \text{ for all } 1 \leq j \leq M \right\}.
\]

Under a given policy \( \Pi \), the total costs \( k^{\Pi}(H) \) over a lifetime \( H \) are given by

\[
k^{\Pi}(H) = S^{\Pi}c_d + Nc_0 + \sum_{i=1}^{N+1} C(T_i).
\]

Note that \( N, T_1, \ldots, T_N \) are the decision variables in our model. The objective is to minimize the total costs, which we denote as

\[
k^*(H) = \min_{\Pi} k^{\Pi}(H),
\]

with corresponding optimal policy \( \Pi^* \) specified by the number of upgrades \( N^* \) and the interupgrade times \( T_i^* \).

If \( c_d = 0 \), we change the total costs variable from \( \mathcal{K} \) to \( \mathcal{C} \) to stress the fact we consider the base case. Under a given policy \( \Pi \), the total costs reduce to

\[
C^{\Pi}(H) = Nc_0 + \sum_{i=1}^{N+1} C(T_i).
\]

We denote \( C^*(H) \) as the minimal total costs in the base case.

A central question is whether we can derive results on what the optimal policy looks like structurally. A straightforward observation is that it can never be optimal to upgrade more than a finite number of times.

**Lemma 1.** Let \( \bar{N} = (\mathcal{C}(H) + v(0))/(c_0 - v(0)) \in [0, \infty) \). Any policy \( \Pi \) with \( N > \bar{N} \) upgrades satisfies \( k^{\Pi}(H) > k^*(H) \).

A short proof of Lemma 1 can be found in Appendix C. As a consequence, Lemma 1 indicates certain scenarios where it is optimal to never upgrade a system during the asset’s lifetime. This typically applies to expensive
systems with little technological advancements and that are robust to failures.

**Corollary 1.** If the upgrade costs are very high, it is never optimal to upgrade. Specifically, this is the case if \( c_0 - 2\nu(0) \geq C(H) \).

Next to the upgrade costs, we can identify two extreme cases with respect to the penalty \( c_d \). The first extreme case is when an upgrade disrupts the asset’s operation completely and an extremely high penalty \( c_d \) is incurred. In that case, it would only be rewarding to upgrade during overhauls, which reduces the number of possible upgrade plans significantly. One needs to only decide at the \( m \) overhaul moments whether to upgrade or not, where \( m \) is finite and relatively small. This can be solved through dynamic programming, which we discuss in more detail in Section 5.2.

The other extreme case concerns the base case where \( c_d = 0 \)—that is, no penalty is given to doing upgrades if it does not coincide with an overhaul. This corresponds to systems that can be easily maintained and upgraded without affecting the total operation mode of the asset. It turns out that the optimal policy has a simple structure for a large class of cycle-cost functions. We consider this in more detail next.

## 4. Analysis of the Base Case

In this section, we assume that \( c_d = 0 \)—that is, the presence of overhauls for the asset is not relevant—and determine the optimal upgrade policy for different shapes of the cycle-cost function. We use these results as building blocks to design an efficient algorithm to find the optimal policy for the general case where also positive values of the penalty \( c_d \) are allowed. Therefore, we spend significant effort to determine the structure of the optimal upgrade policy in the base case. The proofs of the results in this section can be found in Appendix C.1.

Recall that the total cost expression simplifies to (3.5) in the base case for a given policy \( \Pi \). In view of Lemma 1, we can try to derive for every \( N \in \mathbb{N}_{\geq 0} \) with \( N < \bar{N} \) what the corresponding optimal interupgrade times are. If we can answer this question, we can simply compare the total costs for a finite number of upgrades to derive the optimal upgrade policy.

Mathematically, this means that our problem can be rewritten as

\[
C'(H) = \min_{0 \leq N \leq \bar{N}} \left\{ NC_0 + \min_{T_i + \cdots + T_N = H} \left\{ \sum_{i=1}^{N+1} C(T_i) \right\} \right\}.
\]

(4.1)

For a fixed \( N \), we can determine the optimal upgrade policy by deriving the interupgrade times that minimizes

\[
\sum_{i=1}^{N} C(T_i) + C(H - \sum_{i=1}^{N} T_i),
\]

(4.2)

where \( T_i \in [0,H] \) for all \( i = 1, \ldots, N \). Over the finite interval \([0,H]\) and for a fixed \( N \), a minimum of (4.2) can only be attained at points where each partial derivative (in \( T_i \)) is either zero or at the boundaries. However, we know that the latter is not possible, as the system upgrade costs more than salvaging the system immediately (i.e., for the optimal upgrade policy, it cannot hold that \( T_i = 0 \) for some \( i = 1, \ldots, N \) or \( \sum_{i=1}^{N} T_i = H \) because \( c_0 > \nu(0) \) by Assumption 1). Therefore, an optimal upgrade policy must satisfy the property that every partial derivative (in \( T_i \)) is zero. That is, a necessary condition for optimality is given by

\[
C'(T_i) = C' \left( H - \sum_{j=1}^{N} T_j \right), \quad \forall i = 1, \ldots, N',
\]

(4.3)

where \( C'(t) \) denotes the first derivative of the cycle-cost function \( C(t), t \geq 0 \). To find the optimal upgrade policy, we are therefore interested in the sets

\[
T^N = \left\{ (t_1, t_2, \ldots, t_N) \in \mathbb{R}_{\geq 0}^N : \sum_{j=1}^{N} t_j < H, \right\}
\]

\[
C'(t_i) = C' \left( H - \sum_{j=1}^{N} t_j \right), \quad i = 1, \ldots, N \right\}, \quad N \geq 1.
\]

(4.4)

These sets represent the candidates for the optimal interupgrade times if \( N \geq 1 \) upgrades would be executed during the lifetime. The optimal policy is therefore contained in the set

\[
T = \bigcup_{N=0}^{\bar{N}} T^N,
\]

(4.5)

where \( T^0 = \emptyset \) represents the case where no upgrades are executed.

The size of the sets \( T^N \) depends on the behavior of the cycle-cost function \( C(\cdot) \). When we take a closer look at our assumptions, we observe that more information can be derived on the behavior of the cycle costs. Recall that the cycle costs are given by (3.2), and the first derivative is thus given by

\[
C'(T) = -\nu'(T) + c_f(T) + k(T)l(T) \geq 0.
\]

Assumptions 2 and 3 imply that \( c_f(T) + k(T)l(T) \) is non-negative and nondecreasing in \( T \), and, hence, this gives rise to a convex term in the cycle-cost function. This means that the cycle-cost function \( C(\cdot) \) is an aggregation of a convex nondecreasing function and a general non-decreasing function \(-\nu(T), T \geq 0 \). Naturally, the salvage-value function \( \nu(\cdot) \) can take different forms, leading to a certain behavior for the cycle-cost function \( C(\cdot) \). In our analysis, we first focus on the special cases where the cycle costs are convex (Section 4.1), concave (Section 4.2), or S-shaped (Section 4.3). By building on these results, we then consider a generalized structure for the cycle-
cost function and establish the optimal upgrade policy structure in Section 4.4.

4.1. Convex Cycle Costs

In this section, we consider the case where the salvage values are such that the cycle costs are convex and nondecreasing. In practice, there are many real-life scenarios that fall under this case. An example is the case with a constant salvage value, which occurs when a system cannot be resold after use or when the disposal of an older version comes with some cost that is not affected by time. This setting leads to convex cycle costs. In some other settings, where reselling is an option, initially, the resell value can be close to the initial price, but as newer technologies come to the market, the resell value decreases rapidly at a nondecreasing speed (i.e., the speed of reduction in the resell value can be constant, leading to a linearly decreasing salvage value; or it can be increasing, leading to a concave salvage value). In general, we can say a nonincreasing concave salvage-value function \( v(t) \) leads to convex nondecreasing cycle costs. Even if the decrease in the salvage value slows down over time, as long as the moment at which this slowing down occurs before the end of the lifetime of the asset, the cycle-cost function continues to be a convex and nondecreasing function.

If the cycle costs are convex and nondecreasing, it turns out that an optimal upgrade policy for the base model has a simple structure. Specifically, an optimal strategy to follow is to either never upgrade or to upgrade a finite number of times \( N^* \) with equidistant interupgrade times \( T_i = H/(N^* + 1) \), \( i = 1, \ldots, N^* \). Intuitively, it is clear that it is never optimal to upgrade more than a finite number of times because every upgrade always comes with a certain strictly positive cost. Equidistant interupgrade times is a property that comes from the convexity of the cycle costs. This can be illustrated with a simple example as follows. Suppose that there will only be one upgrade during the asset’s lifetime, and this upgrade takes place at time \( T \). Then, the total cost is given by \( c_0 + C(T) + C(H - T) \). The cycle costs are convex, and, by definition, this implies that for every \( \Delta \in [0, H/2] \),

\[
0 \leq c_0 + C\left(\frac{H/2 - \Delta}{2}\right) + C\left(\frac{H/2 + \Delta}{2}\right) \leq c_0 + 2C\left(\frac{H/2}{2}\right)
\]

where an equality holds if \( \Delta = 0 \). Therefore, the total cost is minimized if \( T = H/2 \). Similarly, this also holds in general when executing multiple upgrades, as formalized in Proposition 1.

**Proposition 1.** Suppose that the cycle costs are convex. Then, it is either optimal to never upgrade, or to do so for \( N^* \in \mathbb{N} \) times with \( N^* \leq N \). In the latter case, \( T_i = H/(N^* + 1) \), \( i = 1, \ldots, N^* \), and the minimal total costs are given by

\[
C^*(H) = \min_{N \in \{0, 1, \ldots, [N]\}} \left\{ NC_0 + (N + 1)C\left(\frac{H}{N + 1}\right) \right\}. 
\]

Note that \( N \) is the upper bound on the optimal number of upgrades, as characterized in Lemma 1. Proposition 1 shows that the problem of finding the optimal upgrade times is equivalent to the problem of finding the optimal number of (equidistant) upgrades. That is, the optimal upgrade policy can be found in practice by identifying the value of \( N \in \{0, 1, \ldots, [N]\} \) that minimizes \( C^N(H) \) for a given asset lifetime \( H \), where

\[
C^N(H) = NC_0 + (N + 1)C\left(\frac{H}{N + 1}\right). 
\]

In case that the salvage value of a new system is close to the upgrade price (i.e., \( c_0 - v(0) \) is small), the value of \( N \) can be high. Thus, it may be necessary to check a large number of possible \( N \) values to be able to confirm the optimality. In Lemma 2, we show that the function \( C^N(H) \) has a discrete convex structure in \( N \) for a given asset lifetime \( H \).

**Lemma 2.** If the cycle-cost function is convex, then the function \( C^N(H) \) is discrete convex in \( N \) for a given \( H \).

For an asset with lifetime \( H \), the discrete convex structure of \( C^N(H) \) in \( N \) can be exploited to speed up the process of searching for the optimal number of upgrades \( N^* \) by providing an optimality guarantee. To be specific, Proposition 1 and Lemma 2 imply that, in order to find an optimal number of upgrades \( N^* \), we need to determine the final integer \( N \) before \( C^N(H) \) increases.

**Example 1.** Consider two settings, where in both settings, \( H = 30 \) years and \( c_0 = 4 \). In setting A, let the cycle costs be given by

\[
C_A(t) = \frac{t}{3} + \frac{3}{16} \left(\frac{t}{3}\right)^2 + \frac{1}{10} t^{1.1}.
\]

and in setting B, let

\[
C_B(t) = \frac{t}{3} + \frac{3}{16} \left(\frac{t}{3}\right)^3 + \frac{1}{10} t^{1.1}.
\]

In Figure 2, we plot the cycle costs as a function of time.

Note that the cycle costs are strictly convex on \([0, H]\) for both settings. In view of Proposition 1 and Lemma 2, we can conclude that the (unique) optimal policy is to upgrade \( N^* \) times after every \( H/(N^* + 1) \) years, where \( N^* \) is the final integer before \( C^N(H) \) increases. Using Table 2, we observe that it is optimal to execute an upgrade once \((N_A^* = 1)\) after 15 years in setting A. In setting B, it is optimal to upgrade \( N_B^* = 4 \) times after every six years. \( \Box \)
4.2. Concave Cycle Costs

In this section, we consider the case where the salvage values are such that the cycle costs are concave and non-decreasing. In practice, there can be settings where the salvage value of a new system rapidly decreases as soon as it is installed to the asset, and the speed of decrease in the salvage value can slow down over time (this is the direct opposite of the case in Section 4.1), leading to a convex nonincreasing salvage-value function \( v(t) \). Recall that the first derivative of the cycle-cost function \( C(t) \) is given by \( C'(t) = -v'(t) + c_f(t) + k(t)h(t) \). That is, when the rate of increase in the costs associated with the functionality gap and the component failures over time (captured by the term \( c_f(t) + k(t)h(t) \)) is small relative to the rate of loss in the salvage value, we observe concave cycle-cost functions. For example, consider a system that has a convex nonincreasing salvage value with no functionality gap penalty (i.e., \( c_f(t) = 0 \)). This may be the case when the operator of the asset only requires an operating system, but no more than that, even if the technology increases and there are more efficient systems in the market. If such a system has components that fail at a constant rate (e.g., electronic components) and can be repaired at similar costs, regardless of the age of the system, we end up with concave cycle costs.

For concave cycle-cost functions, it is optimal to never upgrade. Intuitively, this statement can be explained as follows. If it would be optimal to execute at least one upgrade during the lifetime, then, due to the concavity of the cycle-cost function, an optimal strategy would be to immediately execute all upgrades at the start of the asset’s lifetime. But because an upgrade always comes with a positive cost, this implies that it is best to never upgrade at all. We formalize this result in Lemma 3.

**Lemma 3.** Suppose that \( C(t) \) is concave for all \( t \in [0, H] \). Then, it is optimal to never upgrade.

| Table 2. Total Costs Values for Setting A and B |
|---|---|---|
| \( N \) | \( C_A^N(H) \) | \( C_B^N(H) \) |
| 0 | 32.9653 | 201.7153 |
| 1 | 27.3081 | 64.8081 |
| 2 | 28.0268 | 42.6101 |
| 3 | 30.3572 | 37.3884 |
| 4 | 33.3387 | 37.0887 |
| 5 | 36.6489 | 38.7322 |

4.3. S-Shaped Cycle Costs

Section 4.1 considers convex and nondecreasing cycle-cost functions, where the increase in the costs grows over time. Afterward, Section 4.2 considers concave and nondecreasing cycle-cost functions, where the increase in the costs gets slower over time. Although there are realistic settings where each behavior is plausible, there are also cases where the cycle-cost function shows a combination of these behaviors. Suppose that the increase in cycle cost grows fast at the early phases of the life of the system, then the growth stabilizes, and, finally, the growth becomes slower. This behavior for the cycle cost is referred to as an S-shaped function.

In this section, we consider the case that the cycle-cost function is S-shaped—that is, there exists a point of inflection \( x \in [0, H] \) such that the cycle costs are convex on the interval \([0, x]\) and concave on \([x, H]\).

For a typical scenario where S-shaped cycle costs may emerge, one can think of an electronic device, for which upgraded versions with technological improvements appear on the market over time, making the older version less valuable. The salvage value initially decreases slowly, but the drop increases as newer versions appear. This effect is followed by a period where the drop in the salvage value gets smaller again. In other words, the salvage-value function of the electronic device has a reversed S-shape. Depending on the behaviors of the missing functionality penalty and expected failure costs, this, in turn, may result in S-shaped cycle costs.

**Remark 2.** The point of inflection \( x \in [0, H] \) does not necessarily have to be uniquely defined. For example, if the cycle costs are linearly increasing on \([a, b] \subset [0, H] \) with \( a < b \), strictly convex on \([0, a] \), and strictly concave on \([b, H] \), then any \( x \in [a, b] \) can be chosen as the point of inflection. Yet, in the results that are provided in this section, we require the point of inflection \( x \) to be chosen in a (unique) specific manner. That is, we say that \( x \) is chosen such that

\[
C(x + \Delta) - C(x) < C'(x)\Delta, \quad \forall \Delta > 0. \tag{4.10}
\]

For the mentioned example, this implies that the point of inflection is chosen to be equal to \( b \). In the remainder of this paper, points of inflection that satisfy this
Property (4.10) will be referred to as satisfying the technical requirement.

It turns out that in the case of S-shaped cycle costs, there is an optimal policy that has a relatively simple structure. It has one of the following structures:

1. It is optimal to never upgrade;
2. It is optimal to upgrade $N^* \geq 1$ times after every $H/(N^* + 1)$ time units;
3. It is optimal to upgrade $N^* \geq 1$ times after every $T < H/(N^* + 1)$ time units, where $T$ differs from the time between the final upgrade and the end of the lifetime.

The reason that this holds already follows implicitly from the results in the two previous sections. Suppose there is an optimal policy where we upgrade $N^* = N_1 + N_2 - 1$ times, where $N_1$ and $N_2$ are the number of times we have interupgrade times in the interval $[0, x]$ and $(x, H]$, respectively (where we slightly abuse definitions by including the time between the last upgrade and the end of the lifetime as an interupgrade time). Because the cycle costs are concave on $(x, H]$, it follows from our analysis in Section 4.2 that whenever $N_2 \geq 2$, there exists an optimal policy where at least one of the $N_2$ interupgrade times lies at the boundary of the interval $(x, H]$. This is not possible, and, therefore, it must hold that $N_2 \leq 1$. In particular, if $N_2 = 1$, without loss of generality, we can set the corresponding interupgrade time as the time between the final upgrade and the end of the lifetime—that is, as $T_{N_{i+1}}$. Moreover, it follows from our analysis in Section 4.1 that there exists an optimal upgrade policy where the $N_i$ interupgrade times in interval $[0, x]$ equal one another (if any) because the cycle costs are convex on this interval.

In order to prove this formally, we require some additional results on the optimal policy structure; see Appendix C for details. To be specific, we first show in Lemma C.1 that, if $N^* \geq 1$, there exists an optimal policy where no interupgrade time lies strictly between $\min\{x, H/(N^* + 1)\}$ and $\max\{x, H/(N^* + 1)\}$. In Lemma C.2, we then show that there exists an optimal policy that always contains at least one interupgrade time that is at most equal to the point of inflection $x$ of the cycle-cost function—that is, $T_i \leq x$ for some $i = 1, \ldots, N^* + 1$. By using these results, Proposition 2 shows the main result of this section, which gives insight into the structure of the optimal upgrade policy.

**Proposition 2.** Let the cycle costs have an S-shape with point of inflection $x \in (0, H)$ satisfying the technical requirement. It is either optimal to execute no upgrades, or there exists an optimal policy with $N^* \geq 1$ upgrades that has one of two structures: either $T_1 = \cdots = T_N = H/(N^* + 1) = T_{N^*+1}$ or $T_1 = \cdots = T_N \leq x < T_{N^*+1}$.

Besides providing an analytical insight on the structure of the optimal upgrade policy, Proposition 2 is useful, as it establishes that the set of optimal upgrade policies can be simplified to the union of the set $T^0$ (no upgrading) and the sets

$$T_{N_1, N_2} = \{(t_1, t_2) \in \mathbb{R}^2 : 0 < t_1 \leq x < t_2 < H, N_1 \in \mathbb{N}, N_2 \in \{0, 1\}, N_1 t_1 + N_2 t_2 = H, C(t_1) = C(t_2)\},$$

(4.11)

where $N_i$ is bounded due to Lemma 1. To illustrate how this result can be used to determine an optimal upgrade policy, we consider the following example.

**Example 2.** Suppose that $H = 30, c_0 \geq 1, c_i(t) + k_i(t) h(i) = 0$, and

$$C(t) = -v(t) = -1 + (1 + \exp^{-(t-10)})^{-1}.$$  

This is an S-shaped function with $C(t) = -v(t) \in (-1,0)$ for all $t \in [0,H]$ and (unique) point of inflection $x = 10$. Note that

$$C'(t) = -v'(t) = \exp^{-(t-10)}(1 + \exp^{-(t-10)})^{-2}.$$  

This derivative satisfies $C'(t) = C'(20 - t)$ for all $t \in [0,10]$ and is strictly decreasing for $t \geq 10$. In particular, this implies that there exists no optimal upgrade policy where $T_i \in (20,30)$ for some $i = 1, \ldots, N + 1$. In view of (4.11), we observe that for any $t_2 = 20 - t_1$ with $t_1 \in [0,10]$ and $N_2 = 1$,

$$N_1 t_1 + N_2 t_2 = H \implies t_1 = \frac{10}{N_1 - 1} = \frac{10}{N - 1}.$$  

Therefore, the possible candidates for the optimal upgrade policy are $T^0$ and

$$T_{N_1,0} = \left\{\left(\frac{30}{N + 1}, 0\right)\right\}, \quad N \in \mathbb{N}_{\geq 2}$$

$$T_{N_1,1} = \left\{\left(\frac{10}{N - 1}, 20 - \frac{10}{N - 1}\right)\right\}, \quad N \in \mathbb{N}_{\geq 2}.$$  

We point out that any policy $\Pi \in T_{N_1,1}$ leads to total costs that satisfy $c^{\Pi}(H) \geq 0 > C(H)$ for every $c_0 \geq 1$—that is, every policy $\Pi \in T_{N_1,1}$ leads to a higher total costs than when upgrades would never be executed, and, hence, the optimal upgrade policy will never be contained in the set $T_{N_1,1}$. Therefore, depending on the value of $c_0$, the optimal policy is to either never upgrade or to upgrade $N \geq 2$ times with equidistant upgrade times that equal the remaining lifetime (i.e., the time between the final upgrade and the end of the lifetime). \(\square\)

As shown in Example 2, equidistant upgrade times can be optimal under an S-shaped cycle-cost function. In Figure 3, we illustrate the structure of the policy that upgrades three times (i.e., at every 7.5 time units) for the instance introduced in Example 2. Note that the costs of the corresponding policy can be easily calculated, and this can be repeated for other $N$ values to identify the
optimal upgrade policy. Section 5.1 will introduce an efficient solution algorithm to identify the optimal number of upgrades \( N^* \) by exploiting the analytical results established so far.

To illustrate that it is possible to have a (unique) upgrade policy where the remaining time between the last upgrade and the end of the lifetime differs from the (other) interupgrade times, we provide the following example.

**Example 3.** Consider a setting where \( H = 10 \) years, and the system is not subjective to failure—that is, \( h(t) = 0 \) for all \( t \geq 0 \). Initially, the salvage value is given by \( v(0) = 0.15 \). After \( s = 4.9 \), an upgrade comes to the market against upgrade costs \( c_0 = 0.75 \). This event causes the missing functionality penalty to jump to \( c_f(t) = 0.15 \) for all \( t \geq 4.9 \) (while \( c_f(t) = 0 \) for all \( t < 4.9 \)). Moreover, this event leads to the salvage value to decrease rapidly until time \( t = 5 \), after which reselling the old system yields no return. More specifically, the salvage value is given by

\[
v(t) = \begin{cases} 
0.15 & \text{if } t \leq 4.9, \\
0.15 - 30(t - 4.9)^2 & \text{if } 4.9 \leq t \leq 4.95, \\
30(5-t)^2 & \text{if } 4.95 \leq t \leq 5, \\
0 & \text{if } t \geq 5.
\end{cases}
\]  

This leads to an S-shaped continuous cycle-cost function, given by

\[
C(t) = \begin{cases} 
-0.15 & \text{if } t \leq 4.9, \\
-0.15 + 30(t - 4.9)^2 + 0.15(t - 4.9) & \text{if } 4.9 \leq t \leq 4.95, \\
-30(5 - t)^2 + 0.15(t - 4.9) & \text{if } 4.95 \leq t \leq 5, \\
0.15(t - 4.9) & \text{if } t \geq 5.
\end{cases}
\]  

The point of inflection is thus given by \( x = 4.95 \). This cycle-cost function is illustrated in Figure 4.

To determine the optimal upgrade policy, note that \( N^* = (C(H) + v(0))/(c_0 - v(0)) = 1.525 \), and, hence, it is never optimal to upgrade more than once. Because \( C'(t) = 0.15 \) for all \( t \geq 5 \) and \( C'(t) > 0.15 \) for all \( t \in (4.9, 5) \), it is never optimal to upgrade at a time in the interval \((4.9, 5)\) by (4.3). Moreover, because the failure rate equals zero, there is no advantage to upgrading before \( t = 4.9 \). In view of Proposition 2, the optimal policy is therefore either to never upgrade, or to upgrade once at time \( T_1 = 4.9 \) or at time \( T_1 = 5 \), with corresponding total costs \( C(10) = 0.765, c_0 + C(4.9) + C(5.1) = 0.63 \) and \( c_0 + 2C(5) = 0.78 \), respectively. By comparing these total costs, it is clear that it is optimal to upgrade at time \( T_1 = 4.9 \) years, and the remaining time between this upgrade and the end of the lifetime is 5.1 years. This optimal upgrade policy is illustrated in Figure 5. □

**4.4. Generalized Cycle Costs**

In previous sections, we established the structure of the optimal upgrade policy when the cycle-cost function (which must be nondecreasing) is convex, concave, or S-shaped. The question is: What if the cycle-cost function does not fall in this class of functions? This can happen in practice, and we address this generalized case in this section. To be specific, a necessary condition for optimality is given by (4.3), and this condition (which assumes the differentiability of the cycle cost \( C(t) \) in \( t \in [0, H] \)) can be exploited to generalize the results of the previous sections. More specifically, one can divide
the lifetime of the system \([0,H]\) into intervals for which the cycle costs are convex and concave. That is, suppose that \([0,H] = \mathcal{H}_1 \cup \ldots \cup \mathcal{H}_k\) for some \(k \in \mathbb{N}\), where \(k\) is finite, where the cycle costs are convex on intervals \(\mathcal{H}_i\) if \(l\) is odd and concave on intervals \(\mathcal{H}_i\) if \(l\) is even, and \(\mathcal{H}_i \cap \mathcal{H}_j = \emptyset\) if \(i \neq j\). In Figure 6, we illustrate such a partition of a generalized cycle-cost function. Note that these intervals can be constructed even in the extreme case when the cycle cost is convex by choosing \(k = 1\) with \(\mathcal{H}_1 = [0,H]\) and when the cycle cost is concave by choosing \(k = 2\) with \(\mathcal{H}_1 = \{0\}, \mathcal{H}_2 = (0,H]\).

Our next result requires the technical condition that the proposed partition of the cycle-cost function can be done in finite \(k\). We point out that \(k\) is typically finite for instances that we observe in practice—that is, the cycle costs do not behave erratically by changing from convex to concave infinitely often on the finite interval \([0,H]\). This is an intuitive observation: Recall that we showed the cycle-cost function \(C(t)\) is an aggregation of a convex nondecreasing function and a nondecreasing function \(-v(T_i), t \geq 0\). Thus, it can be argued that the cycle costs do not behave erratically for practically reasonable forms of salvage-value function \(v(\cdot)\), leading to relatively small \(k\) values. Under the proposed partition of the cycle-cost function, we next provide a characterization of the optimal upgrade policy.

**Proposition 3.** Suppose that the cycle-cost function can be partitioned as proposed in \(k\) intervals with \(k < \infty\). Then, there is always an optimal upgrade policy that satisfies the following properties:

i. For every even \(l\), there is at most one \(i \leq N^*\) for which \(T_i^* \in \mathcal{H}_i\);  

ii. If \(T_i^*, T_j^* \in \mathcal{H}_l\) for some odd \(l\), then \(T_i^* = T_j^*\).

Note that Proposition 3 is a generalization of the results in the previous sections. In case that the cycle costs are convex on \([0,H]\), then property (ii) implies equidistant upgrade times. In case that the cycle costs are concave on \([0,H]\), property (i) implies that there can only be a single \(T_i^* \in [0,H]\), and, hence, it is optimal to not upgrade at all. Finally, if the cycle costs have an S-shape, then properties (i) and (ii) imply Proposition 2.

We note that the calculation of the optimal upgrade policy may become cumbersome when the cycle costs behave erratically, changing from convex to concave very often. As long as \(k\) is not extremely large, the optimal upgrade times can be derived relatively efficiently for any given number of upgrades, leading to a solution approach to obtain the optimal upgrade policy for a generalized cycle-cost function, as further discussed in Appendix B.1.

### 5. Solution Approach

The results of the previous section can be used to determine the optimal policy. Note that we identified two extreme cases with respect to the penalty \(c_d\)—namely, the base case where \(c_d = 0\) and an upgrade does not affect the asset’s operation, or the case where \(c_d = \infty\) and upgrades occur only during overhauls. We discuss the solution approach for both extreme cases first, after which we explain how one can combine the approaches to find the optimal policy for any value of \(c_d\).

#### 5.1. Solution Approach for the Base Case

As argued in Section 4.4, the optimal interupgrade times can be derived efficiently for any finite values of \(N\), as long as the cycle-cost functions are relatively well-behaved—that is, do not change between convex and concave too often up to lifetime \(H\). In particular, we considered three special cases in Section 4. If the cycle costs are concave, then the optimal policy is to never upgrade. If the cycle costs are convex, the optimal number of upgrades is the first integer at which \(\Delta C^{N_1}(H) = C^{N_1+1}(H) - C^{N_1}(H)\) is nonnegative, denoted by \(N^*\). The optimal policy is then to upgrade \(N^*\) times after every \(H/(N^* + 1)\) time units. This yields Algorithm B.1, as given in Appendix B.1. Implicitly, this solution approach is used for Example 1 in Section 4.1.

If the cycle costs are S-shaped with point of inflection \(x \in (0,H)\), we would need to check more possibilities, as
there may be an optimal policy where the time between the last upgrade and the asset’s lifetime differs from the other upgrade times. That is, we need to check the total costs under the policy of never upgrading, under the class of policies with equidistant upgrade times, and finally under the class of policies where the remaining lifetime differs from the other upgrade times. No upgrading leads to total costs $C(H)$. Next, Proposition 2 shows that if it is optimal to upgrade at least once, then there exists a policy with optimal upgrade times $T_1 = \cdots = T_N \leq x$. In the class of equidistant upgrade times, it therefore suffices to look at $N \geq H/x - 1$. In that case, $C(t)$ is convex in $t = H/(N + 1)$, and, hence, Lemma 2 follows through—that is, $C^N(H)$ is convex in $N \geq H/x - 1$. This implies that we can adopt a similar approach as for convex cycle-cost function. That is, we determine the first integer $N \geq H/x - 1$ for which $\Delta C^N(H) = C^{N+1}(H) - C^N(H)$ is nonnegative. Finally, for the class of policies with a different remaining lifetime, note that due to the $S$-shape of the cycle costs and Proposition 2, it holds that an optimal policy satisfies $T_{N+1} \in (x, H)$ and $T_i = (H - T_{N+1})/N$ for all $i = 1, \ldots, N$ if $N \geq 1$. Define
\[
\tilde{C}^N(H, t) := C_0 + NC\left(\frac{H - t}{N}\right) + C(t), \quad N \geq 1, \tag{5.1}
\]
the total costs where $N$ upgrades are executed at equidistant time intervals of length $(H - t)/N$, and the remaining lifetime after the last upgrade is given by $t$. By deriving $t = \max\{H/(N + 1), x\}$ that minimizes $\tilde{C}^N(H, t)$, we inherently determine the optimal policy that minimizes the total costs for this class of policies with $N$ upgrades. Checking for a finite number of possible upgrades (see Lemma 1) leads to all potential optimal upgrade policies with a different remaining lifetime. We formalize this solution approach in Algorithm B.2, which is given in Appendix B.1.

**Example 4.** Recall Example 3. The derived solution could also be found by executing Algorithm B.2 in Appendix B.1.

1. The first step is to check the class of equidistant upgrade times. Note that $[H/x - 1] = 2$ and $C^2(H) = 1.65 > 1.05 = C^2(H)$; this part terminates after a single step and returns $N^* = 2$, $T^* = 5$ and $C = 1.05$.

2. The next step shows that $C(10) = 0.765 < 1.05$, and, hence, we update our values to $N^* = 0$, $T^* = 0$ and $C = 0.765$.

3. Note that $[N] = [(C(H) + v(0))/(c_0 - v(0))] = 1$, and, hence, the loop consists only of a single step. To determine the $t^* \in [5, 5.1]$ that minimizes $\tilde{C}^1(H, t)$, we observe
\[
\frac{\partial \tilde{C}^1(H, t)}{\partial t} = C'(H - t) + C(t)
\]
\[
= \begin{cases} 
60t - 299.7 & \text{if } 5 \leq t \leq 5.05, \\
-60t + 306.3 & \text{if } 5.05 \leq t < 5.1, \\
0.15 & \text{if } t = 5.1.
\end{cases}
\]
This function is minimized at $t^* = 5.1$, for which $\tilde{C}^1(H, t^*) = 0.63 < 0.765 = C$. Therefore, we update our values to $C = 0.63$, $N^* = 1$, and $T^* = 4.9$.

4. Finally, the algorithm returns the optimal policy $N^* = 1$, $T_1 = 4.9$, $T_2 = 5.1$ and corresponding minimum costs $C^*(H) = 0.63$. □

When the cycle costs are not convex, concave, or S-shaped, a more involved analysis needs to be performed. As indicated in Section 4.4, this can still be done as long as the number of times that the cycle costs change from convex to concave or vice versa is not too large, which is the case for most practical instances. That is, we need to find an optimal policy under the class of policies that satisfy the two properties indicated in Proposition 3. By exploiting these two properties, a solution approach can be designed in a similar way as we did for the S-shaped function; see Appendix B.1.

### 5.2. Solution Approach When Only Upgrading During Overhauls

The extreme case $c_d = \infty$, where upgrades are only executed during overhauls, reduces the number of possible upgrade plans significantly. During the asset’s lifetime, one needs to decide only at the $m$ overhauls whether to upgrade or not, where $m$ is finite and relatively small in practice. This can be solved via dynamic programming. In essence, we condition on the first occurrence of executing an upgrade at an overhaul, if any. Suppose that this occurs at some time $T \in (0, H)$; then, we split the horizon $H$ in two parts, $[0, T]$ and $(T, H)$. The total costs during the first part are given by $c_0 + C(T)$, whereas the total costs pose an identical problem, except that the time horizon has changed to a smaller value $H - T$. Because $m$ is relatively small, we can efficiently work backward in time to determine the optimal policy. Details of this approach are given in Algorithm B.3 of Appendix B.2.

**Example 5.** Reconsider the example as described in Example 1. For the additional parameters, suppose that $c_d = \infty$ and $M_i = 5$ for $i = 1, \ldots, 6$ in both settings—that is, there is an overhaul every five years. In setting A, recall that the optimal policy in the base case was to upgrade after 15 years. Because this coincides with an overhaul, this policy remains the optimal choice, and, consequently, the corresponding minimal total costs of 27,308 are not affected. For setting B, the optimal policy does change with respect to the base model: Instead of upgrading every five years, we upgrade every five years together with a planned overhaul. This leads to total costs of 38,732. □

### 5.3. On the Optimal Solution for the General Case

To find the optimal policy for the general case, a similar dynamic programming approach can be taken. To determine the optimal policy, we condition on the first
occurrence we execute an upgrade at an overhaul (if any). This cuts the lifetime in two pieces that can be optimized separately. The expected total costs in the first piece can be determined by using a strategy similar to the base model because we conditioned that no upgrade is executed during an overhaul in this interval. The second piece of the interval is an identical problem for the general model, but with a shorter horizon length. Using dynamic programming, we can then efficiently determine the optimal policy.

More specifically, suppose that the first time an upgrade is executed at an overhaul is at time $T = \sum_{i=1}^j M_i$ for some $j \leq m$. In the first interval $[0, T)$, no upgrade is executed at an overhaul moment. This corresponds to the base case, with a slight modification that the price of executing an upgrade comes with an additional penalty $c_d > 0$. That is, for a given policy $\Pi$, let $\hat{C}^\Pi(T)$ denote the total costs in the base case with price $c_0 + c_d$ for an upgrade, horizon length $T$, number of upgrades $\hat{N}$ with corresponding interupgrade times $T_1, \ldots, T_{\hat{N}}$ and $T_{\hat{N}+1} = T - \sum_{i=1}^{\hat{N}} T_i$. Equivalently to (3.5), the total costs are given by

$$\hat{C}^\Pi(T) = \hat{N} \cdot (c_0 + c_d) + \sum_{i=1}^{\hat{N}+1} C(T_i). \quad (5.2)$$

The price of the upgrade at time $T$ itself is given by $c_0$. Finally, we need to consider the total costs for the second interval. However, this poses an identical problem with a shorter horizon. Because $m$ is relatively small, we can therefore use dynamic program efficiently to determine the optimal policy; see Algorithm B.4 in Appendix B.1.

**Example 6.** Reconsider the example as described in Example 1, together with $c_d = 1.5$ and $M_i = 10$ for $i = 1$, 2, 3 in both settings. By executing Algorithm B.4 in Appendix B, we determine the optimal policy, which is given as follows. In case A, it is optimal to execute an upgrade twice after every 10 years— that is, together with the planned overhauls. The associated costs are 28.027. In setting B, it turns out that it is optimal to upgrade once after 10 years during an overhaul. In the remaining 20 years, it is optimal to upgrade two more times at times that do not coincide with overhauls. That is, the optimal policy prescribes to upgrade at times $\{10, 16\frac{2}{3}, 23\frac{1}{3}\}$ years with associated total costs of 41.794.

**6. Effects of the Input Parameter on the Optimal Total Costs**

We observed that in the base case, the behavior of the cycle-cost function determines the structure of the optimal upgrade policy. When a penalty is incurred for upgrading while there is no overhaul, the optimal upgrade policy can be determined by using dynamic programming (that uses the structural results from the base case). The final outcome depends heavily on the values and functions of the different input parameters. In this section, we take a closer look at this notion, both analytically and numerically. We note that computational times of our solution approach are low in all instances in the examples below. In particular, these times are not influenced by the value of $c_d$. The proofs of the analytical results can be found in Appendix C.2.

**6.1. The Effect of the Penalty $c_d$ for Upgrades Outside Overhaul Moments**

In previous sections, we already observed that there is a strong dependence on the penalty $c_d$. Whenever this penalty is very small, the optimal policy is the same as in the base case. To be specific, if the base case prescribes to upgrade only at moments that happen to not coincide with the overhaul moments, then this also holds for all sufficiently small penalty values of $c_d$. If there is a relatively high penalty $c_d$, it would be very costly to execute upgrades outside overhaul moments, and, hence, it would not be optimal to do so. In fact, whenever it is optimal to only execute upgrades during overhauls for a given penalty $c_d$, then the same strategy is optimal for every case with $c_d \geq \bar{c}_d$. This is rather intuitive: Higher penalties strictly increase the total costs for any policy that executes upgrades outside the overhauls, whereas policies that only upgrade during overhauls yield the same total costs. Proposition 4 generalizes this notion and shows that, as the penalty $c_d$ increases, the optimal policy can only change to a policy where the number of upgrades that are not jointly executed with overhauls is smaller.

**Proposition 4.** As the penalty $c_d$ increases, the optimal policy can only change to one where the number of upgrades outside overhauls is reduced.

In particular, if a policy $\Pi'$ prescribes to only upgrade during overhauls, then $I^{\Pi'}_T = 0$. If $\Pi'$ is the optimal policy for some penalty value $\bar{c}_d$, then Proposition 4 implies that the optimal policy also only upgrades during overhauls for all $c_d \geq \bar{c}_d$.

**Corollary 2.** Suppose that for a penalty $\bar{c}_d$, an optimal policy $\Pi'$ prescribes to upgrade only at planned overhauls. For any setting with $c_d \geq \bar{c}_d$ and where the other parameter settings are the same, $\Pi'$ is also an optimal policy.

We would like to point out that Corollary 2 implies scenarios where the optimal upgrade solution is insensitive to the value of $c_d$. Whenever the base case leads to an upgrade plan where the timing of the upgrades coincide with the overhauls, Corollary 2 implies that this is also the optimal strategy for all $c_d > 0$. In particular, if it
is optimal to never upgrade in the base case, then this is also the optimal strategy for positive values of penalty $c_d$.

**Example 7.** To illustrate Proposition 4, we reconsider setting B of Example 1. That is, suppose that the horizon length is given by $H = 30$, the price is $c_0 = 4$, and we have cycle costs

$$C(t) = \frac{t}{3} + \frac{3}{16} \left( \frac{t}{3} \right)^3 + \frac{1}{10} t^{1.1}.$$  

Moreover, two overhauls are planned every 10 years—that is, $M_t = 10$, $i = 1, 2, 3$. We consider how the optimal upgrade policy is influenced by $c_d$. In view of Figure 7, the optimal upgrade policy is:

- If $c_d \in [0, 0.29973)$, upgrade every six years at times $\{6, 12, 18, 24\}$ with $K^*(H) = 16 + 5C(6) + 4c_d$;
- If $c_d \in [0.29973, 1.40559]$, upgrade at times $\{7.5, 15, 22.5\}$ with $K^*(H) = 12 + 4C(7.5) + 3c_d$;
- If $c_d \in [1.40559, 1.90805)$, upgrade during one overhaul moment and two times after $6\frac{2}{3}$ years—that is, at times $\{10, 16\frac{2}{3}, 23\frac{1}{3}\}$ or $\{6\frac{2}{3}, 13\frac{1}{3}, 20\}$ with $K^*(H) = 12 + C(10) + 3C(6\frac{2}{3}) + 2c_d$;
- If $c_d \geq 1.90805$, upgrade twice during the overhauls at times $\{10, 20\}$ with $K^*(H) = 8 + 3C(10)$. □

In Example 7, we observe that, as the penalty of upgrading outside an overhaul moment ($c_d$) is increasing, the number of upgrades outside an overhaul moment is nonincreasing (as already shown in Proposition 4), and the number of upgrades at an overhaul is nondecreasing. Also, we see that the total number of upgrades is a nonincreasing function of $c_d$. This is intuitive, as one can expect that, given everything else is equal, the higher a cost associated with an upgrade, the lower the number of upgrades. However, interestingly, this does not hold in general as illustrated next.

**Example 8.** Suppose that everything is the same as in Example 7, except that the cycle-cost function is now from the setting A of Example 1—that is,

$$C(t) = \frac{t}{3} + \frac{3}{16} \left( \frac{t}{3} \right)^3 + \frac{1}{10} t^{1.1}.$$  

The optimal upgrade policy is then as follows:

- If $c_d \in [0, 0.71872)$, upgrade only once at time (15) with $K^*(H) = 4 + 2C(15) + c_d$;
- If $c_d \geq 0.71872$, upgrade at times $\{10, 20\}$ with $K^*(H) = 8 + 3C(10)$. □

Example 8 shows that, as $c_d$ increases, the scenarios where upgrading occurs during overhauls may become relatively more attractive, leading to more frequent upgrades.

### 6.2. The Effect of the Number of Overhaul Moments

As the number of overhauls increases, it generates more possibilities to execute upgrades without having to pay an additional penalty $c_d$. This does not necessary imply that the number of upgrades (during overhauls) increases together with the number of overhauls. The lack of this monotonicity property can be easily explained as follows. There is an optimal upgrade policy in the base case—that is, the optimal policy to minimize costs if no penalty $c_d$ is incurred. This is attained in the general setting if the overhauls coincide exactly with the upgrade moments in this policy. Therefore, one cannot expect to do better than this, even if more overhauls are planned. To illustrate this notion, we consider the following example.

**Example 9.** Reconsider setting B of Example 1 with $H = 30$, $c_0 = 4$, and $c_d = 5$. In the base case when no penalty is incurred, the optimal policy is to upgrade $n = 4$ times (every six years) with minimal costs of 37.0887. This lower bound on the total costs for the general setting. If the overhauls are planned equidistantly, this implicitly also implies that this optimal solution is obtained whenever the number of overhaul moments is given by $m = 5k - 1$ for some $k \in \mathbb{N}$. Indeed, in Table 3, we observe that the total costs are higher when, for example, $m = 5$ than when $m = 4$. Moreover, note that as the number of overhauls grows from $m = 1$ to $m = 2$, the total number of upgrades decreases.

### 6.3. The Effect of Price $c_0$

Naturally, as the price $c_0$ increases, we can expect fewer upgrades, and, indeed, this intuitive notion turns out to be correct. Note that Assumption 1 states that we only consider values of $c_0$ such that $c_0 > \nu(0)$.

**Lemma 4.** The optimal number of upgrades $N^*$ is nonincreasing in $c_0$.

Next, we provide an example to illustrate this result.
Table 3. Optimal Upgrade Policy and Minimal Costs in Case of Overhaul Moments $M_i = \frac{H}{(m+1)}, i = 1, \ldots, m$

<table>
<thead>
<tr>
<th>$m$</th>
<th>Optimal upgrade policy $\Pi^*$</th>
<th>$K'(H)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>[7.5,15,22.5]</td>
<td>32.3884</td>
</tr>
<tr>
<td>1</td>
<td>[7.5,15,22.5]</td>
<td>47.3884</td>
</tr>
<tr>
<td>2</td>
<td>[10,20]</td>
<td>42.6101</td>
</tr>
<tr>
<td>3</td>
<td>[7.5,15,22.5]</td>
<td>37.3884</td>
</tr>
<tr>
<td>4</td>
<td>[6,12,18,24]</td>
<td>37.0877</td>
</tr>
<tr>
<td>5</td>
<td>[5,10,15,20,25]</td>
<td>38.7322</td>
</tr>
</tbody>
</table>

*Note.* The numbers in bold correspond to moments that coincide with overhauls.

Example 10. Reconsider setting B of Example 1 with $H = 30$, $c_d = 5$, and two overhaul moments after every 10 years. In Table 4, we display the total costs under the optimal upgrade policy under the condition that the number of upgrades $N$ is fixed. We point out that if $c_0 = 0$, then the optimal policy is to upgrade after every five years with $N = 5$, and, hence, we do not consider larger values of $N$ in Table 4. We conclude that:

$$N^* = \begin{cases} 
5 & \text{if } c_0 \in [0,0.2926] \\
2 & \text{if } c_0 \in [0.2926,10.5944] \\
1 & \text{if } c_0 \in [10.5944,135.907] \\
0 & \text{if } c_0 > 135.907,
\end{cases}$$

where the corresponding optimal policies can be found in Table 4.

Although the optimal number of upgrades changes as $c_0$ increases, we observe that there is a rapid change from $N^* = 5$ upgrades to $N^* = 2$ in a relatively short interval ($c_0 \in [0.29,0.30]$). For the values of $c_0$ within this interval, we also note that it matters relatively little which policy (i.e., $N \in \{2,3,4,5\}$) is chosen, as all lead to similar total costs.

6.4. The Effect of the Cycle Costs

A final aspect concerns the effect of the cycle costs, which can be seen as the combined contribution of the salvage value, the missing functionality penalty function, and the failure aspect. Already in the base case, we observed the strong effect of the cycle costs on the structural properties of the optimal upgrade policy. In this section, we focus on whether we can say more about the upgrade policy whenever the cycle costs start to increase rapidly.

Intuitively, we would always upgrade before a point at which the cycle costs increase very strongly. To a certain extent, this can be made rigorous through the following result. Suppose that the cycle costs increase extremely after some time $z \in (0, H)$. If it is already optimal to upgrade multiple times with all interupgrade times occurring before time $z$ in an alternate setting, where we linearize the cycle costs after time $z$—that is, a setting where a less rapid increase in cycle costs is assumed after time $z$—then this must also be optimal for the original setting.

**Lemma 5.** Consider an alternate setting with the same parameter settings, except $\bar{C}(t) = C(z) + (t - z)C'(z)$ for all $t \geq z$ for some $z \in (0, H)$. If the optimal policy $\Pi^*$ for the alternate setting is to upgrade $\tilde{N}^*$ times with $\tilde{T}_i^* \leq z$ for all $i = 1, \ldots, \tilde{N}^* + 1$, then $\Pi^*$ is also the optimal upgrade policy for the original setting.

We point out that Lemma 5 may imply more information about the structure of the optimal upgrade policy. For example, suppose that there exists an alternate setting with the properties as in Lemma 5 for some $z \in (0, H)$, and the cycle costs $C(t)$ are convex on $[0,z]$. By construction, note that $\bar{C}(t)$ is convex on the entire interval $[0,H]$, and, hence, the optimal upgrade policy $\Pi^*$ is one with equidistant upgrade times. In other words, there is an optimal policy where we upgrade after equidistant time intervals that are at most equal to $z$.

To conclude this section, we would like to point out that Lemma 5 does not imply that the number of upgrades always increases as the total cycle-cost curve steepens. In fact, this is actually not true, as we illustrate in the following example.

Example 11. Suppose that $c_d = 0$ (base case), $c_0 = 0.02$, and $H = 0.5$. For the cycle costs, consider the following. In setting A, let the cycle costs be given by $C_A(t) = t + t^2/10 = \int_0^t (1 + x/5) dx$, and for setting B, $C_B(t) = \sqrt{t}$.

We point out that $(1 + x/5) \geq \sqrt{x}$ for all $x \in [0, H]$, and, hence, it also holds that $C_A(t) \geq C_B(t)$ for all $t \in [0, H]$. In particular, we see that the increase in cycle costs is steeper for setting A than for setting B: see Figure 8. Yet, as we see in Table 5, the total costs are minimized if $N_A^* = 0$ for setting A and if $N_B^* = 2$ in setting B. In other words, although the total costs increase is steeper in setting A, the optimal number of upgrades is larger in setting B.

7. Conclusion

This paper considers a continuous-time stochastic model in order to determine an optimal upgrade policy for systems in an asset. This model is novel for combining several aspects that are relatively underexamined in existing literature: functionality gap, a predetermined overhaul plan, age-dependent cost functions, and finite lifetime of the asset. For this model, we analytically characterize the
Figure 8. Upgrade Costs over $[0, t]$ if $t < H$

![Graph showing upgrade costs over time](image)

structure of the optimal upgrade policy. For the base case with no penalty for not upgrading at an overhaul moment, we establish that it is optimal to never upgrade if the cycle costs are concave and to upgrade at equidistant time intervals if the cycle costs are convex. Based on these two results, we further characterize the structure of the optimal policy first for S-shaped cycle costs, and then for generalized cycle costs. We use these analytical results as building blocks to design an efficient solution approach based on dynamic programming when the penalty for executing upgrades outside overhauls is nonnegative.

Naturally, the optimal upgrade policy depends heavily on the input parameters. Many intuitive sensitivity results can be made rigorous. For example, the optimal number of upgrades cannot increase as the upgrade price becomes larger. A more subtle approach needs to be taken for other input parameters: As the number of upgrading at a nonoverhaul moment increases, the optimal upgrade policy can only change to one where the number of upgrades that are not executed jointly with overhauls is less. However, surprisingly, that does not imply that the optimal number of upgrades is a nonincreasing function of this penalty. Also, when the number of overhauls increases, the optimal number of upgrades can decrease. Finally, the number of upgrades also does not necessarily have to be nondecreasing with the steepening of the cycle costs.

Motivated by a real-life project involving an asset owner and system supplier in partnership, we developed a model that can be used to collaboratively plan the system upgrades. However, for certain product categories, the incentives may not yet exist to make the upgrade decisions together. A future research direction can be to study the problem introduced in our paper from a game-theoretical point of view to build a model of strategic interactions of the asset owner and the system supplier. Another future research direction can be to characterize the optimal upgrade policies that are robust against the lifetime extensions of assets. Finally, future research can extend the problem considered in this paper to the upgrade planning of multiple systems in an asset.

Acknowledgments
The authors thank the MARCONI members for their constructive and invaluable input for establishing the model.

Appendix A. Notation

<table>
<thead>
<tr>
<th>Table A.1. Overview of Notation</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H$</td>
<td>Lifetime of the asset</td>
</tr>
<tr>
<td>$c_0$</td>
<td>Price of a system upgrade</td>
</tr>
<tr>
<td>$v(t)$</td>
<td>Salvage value of the current system that has been in use for time $t$</td>
</tr>
<tr>
<td>$c_i(t)$</td>
<td>Penalty for missing functionality of the current system that has been in use for time $t$</td>
</tr>
<tr>
<td>$k_i(t)$</td>
<td>Failure rate of a system that has been in use for time $t$</td>
</tr>
<tr>
<td>$k(t)$</td>
<td>Expected repair costs for a failed system that has been in use for time $t$</td>
</tr>
<tr>
<td>$c_d$</td>
<td>Penalty for executing an upgrade not during an overhaul</td>
</tr>
<tr>
<td>$M$</td>
<td>The number of overhauls</td>
</tr>
<tr>
<td>$M_i$</td>
<td>Time between overhaul $i - 1$ and $i$, with $M_0 = 0$ and $\sum_{i=1}^{\infty} M_i = H$</td>
</tr>
<tr>
<td>$\Pi$</td>
<td>The upgrade policy (specified by the upgrade times)</td>
</tr>
<tr>
<td>$N$</td>
<td>The number of upgrades</td>
</tr>
<tr>
<td>$T_i$</td>
<td>Time between upgrade $i - 1$ and $i$, with $T_0 = 0$ and $\sum_{i=1}^{\infty} T_i = H$</td>
</tr>
<tr>
<td>$C(i)$</td>
<td>Cycle costs as specified in (3.2)</td>
</tr>
<tr>
<td>$S^{1\Pi}$</td>
<td>The number of upgrades not jointly executed with an overhaul under policy $\Pi$</td>
</tr>
<tr>
<td>$K^{1\Pi}(H)$</td>
<td>The total costs over a lifetime $H$ under policy $\Pi$, as given in (3.3)</td>
</tr>
<tr>
<td>$\tilde{C}^{1\Pi}(H)$</td>
<td>The total costs over a lifetime $H$ under policy $\Pi$ in the base case ($c_d = 0$), as given in (3.5)</td>
</tr>
<tr>
<td>$N$</td>
<td>Upper bound for the number of upgrades under the optimal policy, given by $(C(H) + v(0))/(c_0 - v(0))$</td>
</tr>
<tr>
<td>$C^{N}(H)$</td>
<td>The total costs if $N$ upgrades are executed after every $H/(N + 1)$ time, as given in (4.7)</td>
</tr>
<tr>
<td>$T^{N}$</td>
<td>Set of candidate solutions for the optimal interupgrade times with $N$ upgrades; see (4.4)</td>
</tr>
<tr>
<td>$\tilde{C}^{N}(H,t)$</td>
<td>Total costs if we execute $N$ upgrades every $(H - t)/N$ time; see (5.1)</td>
</tr>
<tr>
<td>$K^{1\Pi}(t)$</td>
<td>The total costs in $(H - t, H]$ under policy $\Pi$</td>
</tr>
</tbody>
</table>
Appendix B. Pseudo-Code for Solution Approaches

B.1. Solution Algorithms for the Base Case

We first present Algorithm B.1 to find the optimal number of upgrades in the base case with a convex cycle-cost function. In practice, the optimal number of upgrades is typically not too large—for example, not exceeding 10—because the time between upgrades is often no less than multiple years for an asset with a lifespan of 30 years. So, Algorithm B.1 terminates quickly for realistic instances. Note that a bisection approach could also be used to improve the efficiency of the algorithm.

Algorithm B.1 (Algorithm to Find Optimal Upgrade Policy in Base Case with Convex Cycle-Cost Function)

Require: Values $H$, $c_0$, and functions $v(t)$, $c_j(t)$, $h(t)$ for $t \in [0, H]$.
1: Set $N = 0$;
   while $C^{N+1}(H) < C^N(H)$ do
      $N = N + 1$
   end
2: return $N = N$ and $T_j = H/(N + 1)$, $i = 1, \ldots, N+1$.

Algorithm B.2 finds the optimal number of upgrades in the base case with an S-shaped cycle-cost function.

Algorithm B.2 (Algorithm to Find Optimal Upgrade Policy and Corresponding Minimum Costs in Base Case with S-Shaped Cycle Costs)

Require: Values $H$, $c_0$, and functions $v(t)$, $c_j(t)$, $h(t)$ for $t \in [0, H]$ (with point of inflection $x$).
1: Set $N = \lfloor H/x \rfloor$;
   while $C^{N+1}(H) < C^N(H)$ do
      $N = N + 1$
   end
2: Set $N = N$, $T^* = H/(N + 1)$ and $C = C^N(H)$.
3: if $C(H) < C$ then
4:   Set $S = C(H)$, $N^* = 0$ and $T^* = 0$;
5: end if
6: for $1 \leq N \leq N = \lfloor (C(H) + v(0))/c_0 - v(0) \rfloor$ do
7:   Determine $t^* = \max\{H/(N + 1), x\}, H\}$ that minimizes $C^N(H, t^*)$ as in (5.1);
8:   if $t^* \in (x, H)$ and $C^N(H, t^*) < C$ then
9:      $C = C^N(H, t^*)$, $N^* = N$ and $T^* = (H - t^*)/N^*$;
10: end if
11: end for
12: return $N^*$ and $T_j = T^*$, $i = 1, \ldots, N^*$, $T_{N+1} = H - N^*T^*$ and $C = C^N(H)$.

Finally, we consider generalized cycle-cost functions that can be partitioned as described in Section 4.4 with $k$ intervals and describe how we can (greedily) derive the optimal policy by exploiting several derived properties in our analysis. Suppose that the optimal upgrade policy has $N^*$ interupgrade times in interval $\mathcal{H}_i$, $i = 1, \ldots, k$. Proposition 3 and Lemma 1 show that the number of possibilities for $(N_1^*, \ldots, N_k^*)$ is finite. More specifically, Proposition 3 and Lemma 1 show that there exists an optimal upgrade policy where the number of upgrades falls in the set $\mathcal{N} := \{(n_1, \ldots, n_k) \in \mathbb{N}^k_0 : n_i \in \{0, 1\} \text{ for } i \text{ odd}, n_i \leq N = (C(H) + v(0))/(c_0 - v(0)) \ \forall i = 1, \ldots, k\}$. (B.1)

Moreover, write $\mathcal{A} := \min_{\alpha \in [\mathcal{A}, \mathcal{A}]} \{C^*(t) \geq 0 \text{ and } \mathcal{A} := \max_{\alpha \in [\mathcal{A}, \mathcal{A}]} \{C^*(t) < \infty \}$. Because the cycle-cost function’s derivative exists on $[0, H]$ and is nonnegative, we can define for any $\alpha \in [\mathcal{A}, \mathcal{A}]$,

$$T_\alpha := ((t_1, \ldots, t_i) \in [(0, H) \cup \emptyset]^k : (\text{either } t_i \in \mathcal{H}_i \text{ and } C^*(t_i) = \alpha, \text{ or } t_i = \emptyset) \ \forall i = 1, \ldots, k).$$  

(B.2)

In view of (4.3) and Proposition 3, we observe that there exists an optimal upgrade policy where the derivative of the interupgrade times is given by some $\alpha \in [\mathcal{A}, \mathcal{A}]$ and, hence, is contained in the set $T_\alpha$.

Consequently, a greedy algorithm to find an optimal upgrade policy can be described as follows: For every $(n_1, \ldots, n_k) \in \mathcal{N}$, determine all upgrade policies $(t_1, \ldots, t_k) \in \cup_{\alpha \in [\mathcal{A}, \mathcal{A}]} T_\alpha$ that yield $\sum_{i=1}^k n_i h_i(t, t_0) = H$ (if any). Minimize the total costs under the determined policies, and save the corresponding outcome (i.e., minimized costs and corresponding interupgrade times). Enumerate over all (finite) values of $N$ to establish the overall minimal costs and corresponding upgrade policy. This greedy algorithm can be used to efficiently obtain the optimal upgrade policy as long as $k$ is not too large.

B.2. Solution Algorithms for Positive Penalty Costs

For Algorithm B.3 and Algorithm B.4, we extend the notation for the total costs $K^1(t)$ as the total costs in $(H-t, H]$ under policy $\Pi$. Let $K^\alpha(t)$ denote the minimal costs over all upgrade policies on this interval. Moreover, we denote $\hat{K}^\alpha(t)$ as the minimal total costs in $(H-t, H]$ among the policies where upgrades are only executed during overhaul moments. We point out that the only difference between Algorithm B.3 and Algorithm B.4 is that in steps 1, 3, and 5, the cycle-cost function $C(t)$ is replaced by $\hat{C}(t)$. In other words, we need to find the optimal upgrade policy in the base case with upgrade price $c_0 + c_d$. Section B.2.1 provides the algorithms to do so in case of convex, S-shaped, and generalized cycle-cost functions.

Algorithm B.3 (Algorithm to Find Optimal Upgrade Policy Among the Class Where Upgrades Are Only Executed During Overhaul Moments)

Require: Values $H$, $c_0$, and functions $v(t)$, $c_j(t)$, $k(t)$, and $h(t)$ for $t \in [0, H]$.
1: Initiate $\hat{K}(M_{m+1}) = C(M_{m+1})$ and save $\hat{T}(m+1) = \{M_{m+1}\}$;
2: for $i = m, \ldots, 1$ do
3:   Set $\hat{K}(\sum_{i=1}^{m+1} M_i) = C(\sum_{i=1}^{m+1} M_i)$ and $\hat{T}(l) = \{\sum_{i=1}^{m+1} M_i\}$
4:   for $j = l, \ldots, m$ do
5:      Set $x = C(\sum_{i=1}^{m+1} M_i) + c_0 + \hat{K}(\sum_{i=1}^{m+1} M_i)$;
6:      if $x < \hat{K}(\sum_{i=1}^{m+1} M_i)$ then
7:         Set $\hat{K}(\sum_{i=1}^{m+1} M_i) = x$ and $\hat{T}(l) = \{\sum_{i=1}^{m+1} M_i\} \cup \hat{T}(j+1)$;
8:   end if
9: end for
10: end for
11: return Total costs $\hat{K}(H) = \hat{K}(\sum_{i=1}^{m+1} M_i)$ and upgrade policy $\hat{T}(1)$.

Algorithm B.4 (Algorithm to Find Optimal Upgrade Policy)

Require: Values $H$, $c_0$, $c_d$ and functions $v(t)$, $c_j(t)$, $k(t)$, and $h(t)$ for $t \in [0, H]$.
C.1. Proofs

Proof of Lemma 1. Write $\Pi_0$ as the policy to never upgrade, and note that $K^\Pi_0(H) = C(H)$. For any policy $\Pi$ with $N > \overline{N}$, $K^\Pi(H) \geq Nc_0 + (N + 1)(-v(0)) > N(c_0 - v(0)) - v(0) = C(H) = K^\Pi_0(H) \geq C(H)$. □

C.1. Proofs for the Base Case

Proof of Proposition 1. We need to minimize the total costs by finding the optimal policy that describes the number of upgrades $N$ and corresponding $T_1, \ldots, T_N$ such that

$$C^*(H) = \min_{\Pi} C^\Pi(H). \tag{C.1}$$

Because of Lemma 1, we know that it is never optimal to upgrade more than $\overline{N}$ times. First, we answer the question of what the optimal interupgrade times are if the optimal number of upgrades is given by $N^*$. If $N^* = 0$, then there are no upgrades during the lifetime, and $C^*(H) = C(H)$. Otherwise, $1 \leq N^* \leq \overline{N}$, and the policy also needs to describe the interupgrade times. Then,

$$C^*(H) = N^*c_0 + \min_{T_1, \ldots, T_{N^*+1} \in 2^{\{1, \ldots, \overline{N}\}}} \left\{ \sum_{i=1}^{N^*+1} C(T_i) \right\}.$$ 

The term in the minimization operation is the summation of $N^* + 1$ identical convex nondecreasing functions in $T_i$ under the condition that the sum of the interupgrade times $T_i$ is equal to $H$. By the convexity property, this is minimized if $T_1 = \cdots = T_{N^*+1} = H/(N^* + 1)$. Using this observation together with Lemma 1, Proposition 2 concludes the proof. □

Proof of Lemma 2. To show that $C^N(H)$ is convex in $N$, we require that $\Delta^2C^N(H) \geq 0$. Note that

$$\Delta^2C^N(H) = C^{N+1}(H) - 2C^N(H) + C^N(H) = c_0 + C(H) - (N + 1)C(H/N + 1) - C^N(H),$$

and hence,

$$\Delta^2C^N(H) = \Delta^2C^{N+1}(H) - \Delta^2C^N(H) = (N + 1)C(H) - C(H) - (N + 3)C(H) - C(H).$$

Write

$$y_i = \frac{H}{N + 1} + i\frac{H}{(N + 1)(N + 2)(N + 3)},$$

and note that $y_{N+3} = H/(N + 1)$ and $z_{N+1} = y_0 = H/(N + 2)$. Therefore,

$$\Delta^2C^N(H) = \sum_{i=1}^{N+3} (C(y_i) - C(y_{i+1})) - (N + 3)\sum_{i=1}^{N+1} (C(z_i) - C(z_{i+1})) = \sum_{i=1}^{N+3} ((C(y_i) - C(y_{i+1})) - (C(z_i) - C(z_{i+1}))) \geq 0,$$

where the final inequality follows because every single term within the summations is nonnegative due to the convexity and the nondecreasing property of the cycle-cost function. □

Proof of Lemma 3. Suppose that there is an optimal policy $\Pi$ for which it is optimal to upgrade exactly $N \geq 1$ times. Because $C(t)$ is concave and nondecreasing in $t \geq 0$, the term $\sum_{i=1}^{N+1} C(T_i)$ is minimized if $T_1 = \cdots = T_N = 0$ and $T_{N+1} = H$. However, in that case, it follows that

$$C^\Pi(H) = Nc_0 - v(0) + C(H) > C(H), \tag{C.2}$$

where the latter corresponds to the total costs under the policy to never execute an upgrade. This contradicts the hypothesis. □

Lemma C.1. Let the cycle costs have an S-shape with point of inflection $x \in (0, H]$ that satisfies the technical requirement. If $N^* \geq 1$, then there exists an optimal upgrade policy where for all $i = 1, \ldots, N^* + 1$, it holds that $T_i \in \min\{x, H/(N^* + 1)\}$, $\max\{x, H/(N^* + 1)\}$. □

Proof of Lemma C.1. This lemma only applies to cases where $x \notin H/(N^* + 1)$; otherwise, $\min\{x, H/(N^* + 1)\}$, $\max\{x, H/(N^* + 1)\} = 0$. Therefore, we assume $x \notin H/(N^* + 1)$ for the remainder of the proof. Suppose that there would be an optimal strategy $\Pi$ with an interupgrade time strictly between point of inflection $\min\{x, H/(N^* + 1)\}$ and $\max\{x, H/(N^* + 1)\}$ with $N^* \geq 1$. Because of symmetry, we can assume without loss of generality that

$$T_{N+1} \in \left( \min\{x, H/(N + 1)\}, \max\{x, H/(N + 1)\} \right).$$

Moreover, note that a necessary condition for an policy to be optimal is

$$C^\Pi(T_i) = C^\Pi(T_{N+1}) \quad \forall i = 1, \ldots, N. \tag{C.3}$$

Moreover, let

$$a = \inf\{t \in [0, x] : C(t) = C^*(T_{N+1})\},$$

$$b = \sup\{t \in [0, x] : C(t) = C^*(T_{N+1})\},$$

$$c = \inf\{t \in [x, H] : C^\Pi(t) = C^\Pi(T_{N+1})\},$$

$$d = \sup\{t \in [x, H] : C^\Pi(t) = C^\Pi(T_{N+1})\}.$$ 

We point out that at least $a$ and $b$ or $c$ and $d$ are well-defined (or all). Moreover, if it is well-defined, note that $a = b$ in case...
of strict convexity on $[0, x]$ and $c = d$ in case of strict concavity on $[x, H]$. Because $C(\cdot)$ has an S-shape with point of inflection $x$ that satisfies the technical requirement, we observe that $[a, b]$ and $[c, d]$ contain all solutions of (C.3), and $C(\cdot)$ is linear on both intervals (if they exist). Therefore, without loss of generality, we can say that $T_i \in [a, b]$ for all $i = 1, \ldots, N_1$ and $T_i \in [c, d]$ for all $i = N_1 + 1, \ldots, N_2$ where $N_1 + N_2 = N + 1$. Note that $N_1 = 0$ if $a, b$ do not exist, and, similarly, $N_2 = 0$ if $c, d$ do not exist.

**Case $x < H/(N + 1)$**. Or in other words, $x < T_{N+1} < H/(N + 1)$. In that case $c, d$ are well-defined values, and $T_{N+1} \in [c, d]$ and $N_2 \geq 1$. If $N_2 = 1$, then $N = N_1 \geq 1$ and $a, b$ must also be well-defined. Therefore, $T_i \leq b \leq x < H/(N + 1)$ for all $i = 1, \ldots, N$. Recalling that $T_{N+1} > H/(N + 1)$ yields

$$\sum_{i=1}^{N+1} T_i < \sum_{i=1}^{N+1} H/(N + 1) = H,$$

which is a contradiction. Therefore, we require that $N_2 \geq 2$. In that case, we can construct the policy $\tilde{\Pi}$ where

$$\tilde{T}_i = T_{i}, \quad i = 1, \ldots, N_1, \quad \tilde{T}_N = T_N + T_{N+1} - x, \quad \tilde{T}_{N+1} = x.$$  

Note that $\tilde{T}_i \in (x, H)$. Because the cycle costs are concave due to the technical requirement, $C(T_N) + C(T_{N+1}) > C(\tilde{T}_N) + C(\tilde{T}_{N+1})$, and, hence,

$$c^\dagger(\tilde{H}) = c^\dagger(\tilde{H}) + C(\tilde{T}_N) + C(\tilde{T}_{N+1}) - C(T_N) - C(T_{N+1})$$

then $c^\dagger(\tilde{H}) < c^\dagger(H)$ (C.4). In other words, our original policy $\Pi$ is not optimal, contradicting our hypothesis.

**Case $x > H/(N + 1)$**. Or in other words $H/(N + 1) < T_{N+1} < x$. In that case, $a, b$ are well-defined, and $T_{N+1} \in [a, b]$ and $N_1 \geq 1$. We will show that if this policy $\Pi$ exists, then there is another policy $\tilde{\Pi}$ that yields the same minimal total costs, but has no interupgrade time in $(H/(N + 1), x)$. First, if $T_i \in [a, b]$ for all $i = 1, \ldots, N$, then let policy $\tilde{\Pi}$ be given by $\tilde{T}_i = H/(N + 1)$ for all $i = 1, \ldots, N + 1$. By construction, it holds that $\tilde{c}^\dagger(\tilde{H}) = c^\dagger(H)$ because the cycle costs are linear on $[a, b]$, and, hence, $\tilde{\Pi}$ also is an optimal upgrade policy.

Next, suppose that $T_i \in [a, b]$ for all $i = 1, \ldots, N$, implying directly that $c$ and $d$ are well-defined. Without loss of generality, we can order the interupgrade times such that $T_i \in [c, d]$ for $i = 1, \ldots, I$ for some $I \geq 1$ and $T_i \in [a, b]$ for $i = I + 1, \ldots, N$ (if any). Consider the policy

$$\tilde{T}_i = T_i, \quad i = 1, \ldots, I, \quad \tilde{T}_{N+1} = \frac{\sum_{i=1}^{N+1} T_i}{N - I + 1}, \quad j = I + 1, \ldots, N.$$  

Note that $\tilde{T}_{N+1} \geq a$ and

$$\tilde{T}_{N+1} = \frac{\sum_{i=1}^{N+1} T_i}{N - I + 1} - \frac{H - \sum_{i=1}^{I} T_i}{N - I + 1} < \frac{H - H/(N + 1)}{N - I + 1} = \frac{H}{N + 1} < x.$$  

That is, policy $\tilde{\Pi}$ has no interupgrade time in $(H/(N + 1), x)$, and $\tilde{c}^\dagger(\tilde{H}) = c^\dagger(H)$ because the cycle costs are linear on $[a, b]$. □

**Lemma C.2.** Let the cycle costs have an S-shape with point of inflection $x \in (0, H)$, where $x$ satisfies the technical requirement. If $N' \geq 1$, then there exists an optimal policy for which $T'_i \leq x$ for some $i = 1, \ldots, N + 1$.

**Proof of Lemma C.2.** Suppose that the statement is not true, and $T_1, \ldots, T_{N+1} > x$. In view of Lemma 1, we observe that

$$\sum_{i=1}^{N+1} T_i \geq \sum_{i=1}^{N+1} H/(N + 1) = H,$$

where $\sum_{i=1}^{N+1} T_i = H$ holds if and only if $T_i = \cdots = T_{N+1} = H/(N + 1) > x$. In other words, if $T_1, \ldots, T_{N+1} > x$, then $T_1 = \cdots = T_{N+1} = H/(N + 1) > x$, and the total costs under this policy are given by $N_0 + (N + 1)C(H/(N + 1))$.

Consider the policy $\tilde{\Pi}$, where $\tilde{T}_i = \cdots = \tilde{T}_N = x$ and $\tilde{T}_{N+1} = H - Nx$. Then, because of the technical requirement and the notion that we have a setting where $x < H/(N + 1)$,

$$\tilde{c}^\dagger(\tilde{H}) = N_0c_0 + NC(x) + C(H - nx) < N_0 + (N + 1)C(H/(N + 1)),$$

contradicting the hypothesis. □

**Proof of Proposition 2.** If $N' = 0$, then the proposition holds, and there is nothing to prove. Therefore, let $N' \geq 1$ for the remainder of the proof. That is, suppose there is an optimal policy where we upgrade $N' = N_1 + N_2 - 1 \geq 1$ times, where $N_1$ and $N_2$ are the number of times we have interupgrade times in the interval $[0, x]$ and $(x, H)$, respectively. In view of (4.3), we observe that there exists such an optimal policy that satisfies $T_1 = \cdots = T_{N_1} = \cdots = T_{N_1 + N_2} = x$ (if $N_2 = 0$). We point out that $N_1 \geq 1$ due to Lemma C.2. To conclude the result, what remains to show is that $N_2 \in [0, 1)$. To that purpose, suppose that $N_2 \geq 2$. Consider the policy $\tilde{\Pi}$ with

$$\tilde{T}_i = T_{i}, \quad i = 1, \ldots, N_1$$

$$\tilde{T}_j = T_{N+1}, \quad j = N_1 + 1, \ldots, N' = 1$$

$$\tilde{T}_{N+1} = 2T_{N+1} - x.$$  

Note that this solution is feasible, because

$$\sum_{i=1}^{N_1 + N_2} \tilde{T}_i = N_1T_1 + (N_2 - 2)T_{N+1} + x + (2T_{N+1} - x) = N_1T_1 + N_2T_{N+1} = H.$$  

Because of the technical requirement, we observe that

$$c^\dagger(H) = N'c_0 + N_1C(T_1) + (N_2 - 2)C(T_{N+1}) + C(x) + C(2T_{N+1} - x) < N_0c_0 + N_1C(T_1) + N_2C(T_{N+1}) = c^\dagger(H).$$

This contradicts the hypothesis, proving our proposition. □

**Proof of Proposition 3.** The result directly follows from the same arguments used in the proof of Proposition 2, so a detailed proof is omitted. □

**C.2. Proofs for the Sensitivity Analysis**

**Proof of Proposition 4.** This is a direct consequence of the Total Cost Function (3.3). Note that for any fixed policy $\Pi$, the total cost function is linear in $c_\Pi$ with slope $S^\dagger$. Therefore, as penalty $c_\Pi$ increases, the optimal policy can only change to one with a smaller value of $S^\dagger$. □

**Proof of Lemma 4.** For every policy $\Pi$, the total costs are given by (3.3). Write $P_N$ as the set of all possible upgrade policies with $N$ upgrades. We observe that for every...
N ∈ N, we can determine the upgrade times that minimize the total costs by solving
\[
K^N(H) := \min_{t \in \mathbb{P}_H} \mathbb{E}[c_0 + \sum_{i=1}^{N+1} C(T_i)],
\]
regardless of the value of \(c_0\). In particular,
\[
\kappa^N = \min_{t \in \mathbb{N}} (Nc_0 + K^N(H)).
\]
We point out that
\[
\frac{\partial}{\partial \nu}(Nc_0 + K^N(H)) = N.
\]
That is, it is linear in \(c_0\) with slope \(N\), and \(N^*\) is the argument that minimizes the total costs. In conclusion, the optimal number of upgrades will never decrease. □

**Proof of Lemma 5.** For any general variable \(X\), let \(\tilde{X}\) denote the corresponding one in the alternate setting. For any policy \(\Pi\), it holds that \(K^{\tilde{X}}(H) \geq \tilde{K}^{\tilde{X}}(H)\) because \(\tilde{C}(i) \leq C(i)\) for all \(i \in [0, H]\). Moreover, because \(\tilde{\tau}^*_i \leq \tau^*_i\) for all \(i = 1, \ldots, N^* + 1\), it holds that \(K^{\tilde{X}}(H) = \tilde{K}^{\tilde{X}}(H)\). In conclusion, we obtain
\[
K^{\tilde{X}}(H) = \tilde{K}^{\tilde{X}}(H) \leq \tilde{K}^{\tilde{X}}(H) \leq K^{\tilde{X}}(H),
\]
for any policy \(\Pi\). We can conclude that \(\tilde{\Pi}^*\) is also optimal for the original setting. □

**References**


