MSc Thesis

Preventing Machine Crashes in a Single- and Multi-Unit System using Markov Decision Processes

F.I. Wienk

Supervisors:
Prof. dr. R.J. Boucherie (University of Twente),
G.J. van der Heijden MSc (Urenco)

October, 2024

Department of Applied Mathematics: Operations Research
Faculty of Electrical Engineering, Mathematics and Computer Science

Acknowledgements

This thesis is written to conclude my master's degree in Applied Mathematics at the University of Twente. I've been fortunate enough to be able to do my final project at Urenco in Almelo, an extraordinary company, renowned for its innovative work and supportive culture. I want to thank my colleagues for their constant support, openness, and genuine interest in my work. Their support and friendliness made this experience both productive and enjoyable.

The past eight months have been both demanding and rewarding. I thank my supervisors, Richard from the University of Twente and Gert-Jan from Urenco, for their invaluable guidance and support. Richard for the insightful meetings, and his critical view. Gert-Jan for the informative tours, stimulating discussions and making sure I translated the mathematics back into reality.

Finally, I'm grateful to my family and friends for their constant support. Their unwavering encouragement throughout this thesis and my studies has meant the world to me.

Femke Wienk,
October 2024

Abstract

In this report we formulate an optimal stopping problem for a single centrifuge that is subject to deterioration. Five different deterioration causes are identified and modelled with a discrete-time five-dimensional absorbing Markov chain. The deterioration process ends with failure, which can be a 'crash' or a 'run down'. Crashes are undesirable, because these can also damage neighbouring centrifuges in the system, while run downs have almost no implications. The model is extended to a system of centrifuges, in which the replacement of all non-operating centrifuges is also considered. The optimal policy for both models is proven to have a monotone structure within the partial ordering of the state and action space. The monotone structure is used in the implementation of the policy iteration algorithm to efficiently neglect certain non-optimal policies. Multiple experiments are performed to determine how the cost parameters, the number of centrifuges and the state dimensions affect the resulting policy.

Keywords: Markov Decision Process; Optimal stopping; Single-unit; Multi-unit; Partial ordering; Monotone policies; Predictive maintenance.

Contents

1 Introduction

Our research is focused on maintenance activities of gas centrifuges at Urenco. First, we explain the uranium process to gain a basic understanding of the different parts of a centrifuge, what a centrifuge does and how a system of centrifuges operates (Section 1.1). This is followed by some background information of the company Urenco (Section 1.2), and the motivation of our research (Section 1.3).

1.1 Uranium Enrichment Process

Uranium (U) is a radioactive chemical element. The decay of uranium isotopes varies between 100.000 years and 4.5 billion years, which is a slow decay. In nature, uranium appears mostly as a mixture of the isotopes U238 (99,27%), U235 (0,72%), and U234 (0,006%). The latter is often considered a neglectable amount. Isotope U235 is the only naturally occurring fissile isotope. This means it can sustain a nuclear chain reaction, which makes U235 suitable as fuel for nuclear reactors. But nuclear reactors require a U235 concentration between 3% and 5%. Therefore, the concentration of U235 in naturally mined uranium is increased via a process called enrichment, which can be done by several methods, such as gaseous diffusion, gas centrifugation, or liquid thermal diffusion. Urenco uses gas centrifuging to concentrate the U235 to low-enriched uranium, meaning the concentration is smaller than 20%. To separate U235 from U238 the compound Uranium hexafluoride (UF6) is used. UF6 can be a solid, liquid, or gas, depending on its temperature and pressure, see the phase-diagram in Figure 1. Gas centrifuges operate with material in gas form and UF6 is solid at normal atmospheric pressure, so it must be heated for the enrichment process. [21].

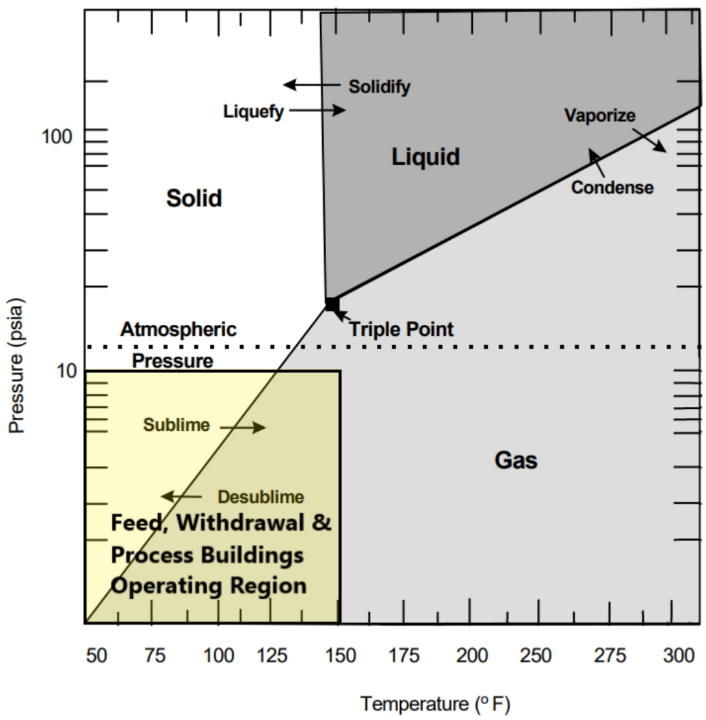


Figure 1: Phase diagram of UF6. Adapted from [34].

A gas centrifuge consists of a cylindrical rotor, casing, and an electric motor. See Figure 2 for a schematic view. The electromagnetic motor rests on the bottom of the casing and turns a shaft attached to the rotor's base. The rotor is centered by the magnetic bearing on top, preventing contact between moving and stationary parts. The rotor is driven by the motor and spins at a high speed (above the speed of sound) inside a vacuum casing. Operating the rotor in a vacuum minimizes the drag, allows for better temperature control, and isolates the machine from vibrations. To achieve such high speeds, each machine is manufactured with close tolerances to minimize the imbalance. Inside the rotor chamber is a rotating disc-shaped baffle and a stationary tube arrangement for feeding and extracting gas. [11, 36]

A centrifuge has three lines for material to travel: feed, product, and tails. The feed is the material entering the centrifuge, while the product and tails are the material exiting it. The product has a higher concentration of U235 than the tails. The feed is fed through a stationary center post. This post also channels the enriched product to the top scoop and the depleted tails to the bottom scoop. The process is balanced, so the amount of material that enters the centrifuge equals the amount leaving it, i.e. *feed = product + tails*. [34, 39, 43].

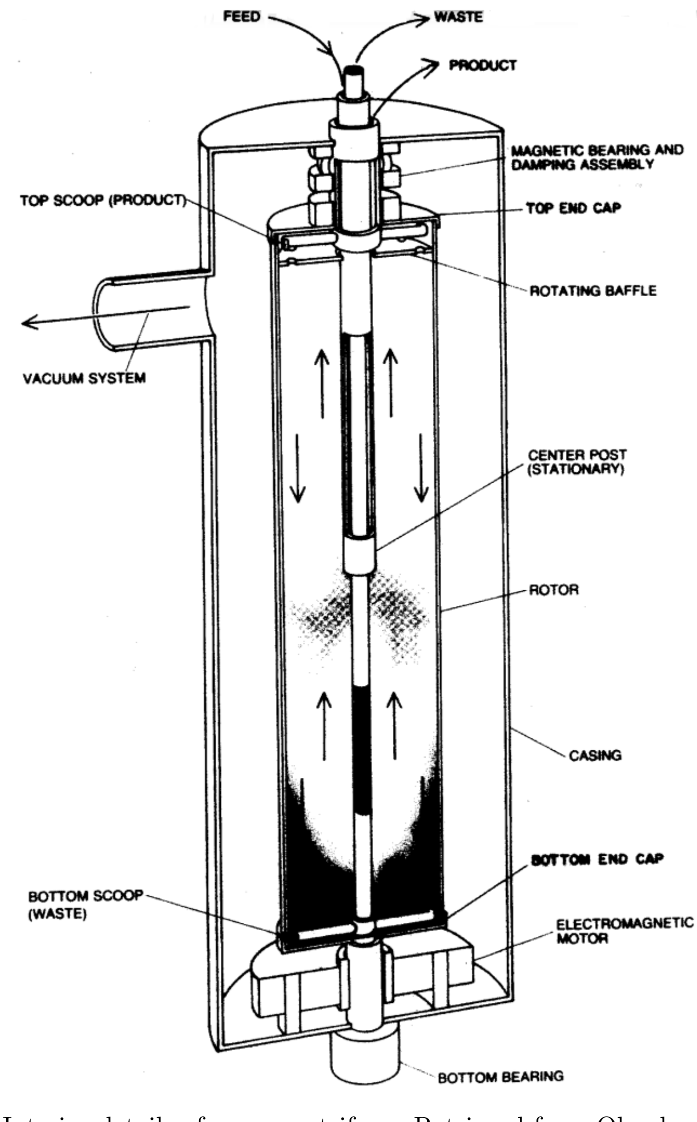


Figure 2: Interior details of a gas centrifuge. Retrieved from Olander (1981) in [39].

Centrifuges use the principle of centripetal force. So lighter molecules (U235F6) are moved to the axis and heavier molecules (U238F6) are forced to the side of the centrifuge. The vertical arrows in Figure 2 indicate the internal gas circulation. The internal circulation is driven by multiple mechanisms. The "thermal drive", is generated by controlling the temperature of the rotor wall and end caps, while the "mechanical drive" results from the interaction between the rotating gas and stationary objects. A single centrifuge cannot achieve the desired level of enrichment. So the process is repeated multiple times to further increase the concentration of U235F6. For this, multiple centrifuges are placed in series and parallel configurations, known as a cascade. The final product of a cascade contains the desired concentration of U235. This product is discharged into a cylinder and solidified for transportation to a nuclear power plant [39].

A cascade can contain over a thousand centrifuges. These centrifuges are of the same type, but may be different versions of that type. This problem focuses on the TC-12 type. Various cascade arrangements with different numbers of centrifuges are possible. Urenco has multiple arrangements in its factory. See Figure 4 for a schematic example of a cascade. Here, centrifuges placed in parallel form a single stage. In reality, a stage consists of multiple flomels, which are groups of centrifuges that receive identical feed and generate the same product and tails. However, there are slight performance differences between the centrifuges in a stage. Whether these differences are negligible depends on the target criteria. Stages are connected in series. The stage where the feed enters and the stages above it are the *enriching stages*, while the stages below are the *depleting stages*. Urenco uses symmetric cascades, where the enriched product from each stage becomes the feed for the next stage, and the tails from each stage are part of the feed of the preceding stage.

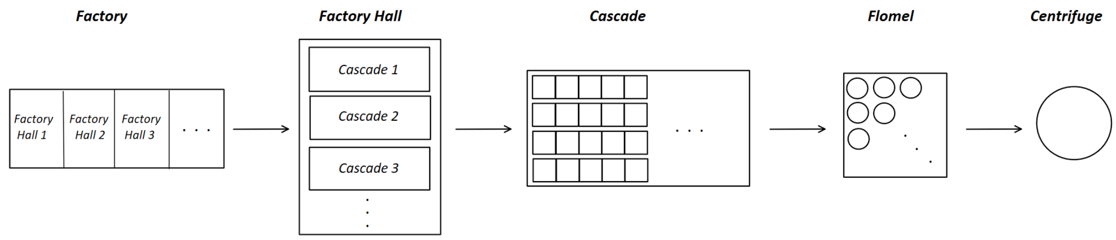


Figure 3: Example of factory layout with the different operational levels.

A flomel has physical and process neighbours. A physical neighbour is a flomel positioned immediately adjacent to another flomel. In the layout depicted in Figure 3, flomels have both vertical and horizontal neighbours, with an aisle separating vertically aligned flomels. Therefore, only the 1 or 2 horizontal adjacent flomels are considered physical neighbours. A process neighbour is a flomel connected to another flomel via pipework according to the different stages as shown in Figure 4. A flomel has 0, 1, or 2 process neighbours.

The enrichment process is a continuous process. So replacing centrifuges requires stopping the whole process. This type of maintenance is known as *refurbishment* and results in a production loss. The capacity of a cascade is defined by the Seperative Work Units (SWU) produced per time unit [37, 34].

Time series data is collected for each centrifuge by the Centrifuge Monitoring System (CMS). The CMS data contains information regarding the power and power-related data of the centrifuge's motor. The performance of a centrifuge is determined by the power data. Next to power, the system measures Cos-Phi data. This is the ratio of the effective power to the apparent power. The system registers when a centrifuge has failed because no power can be measured anymore.

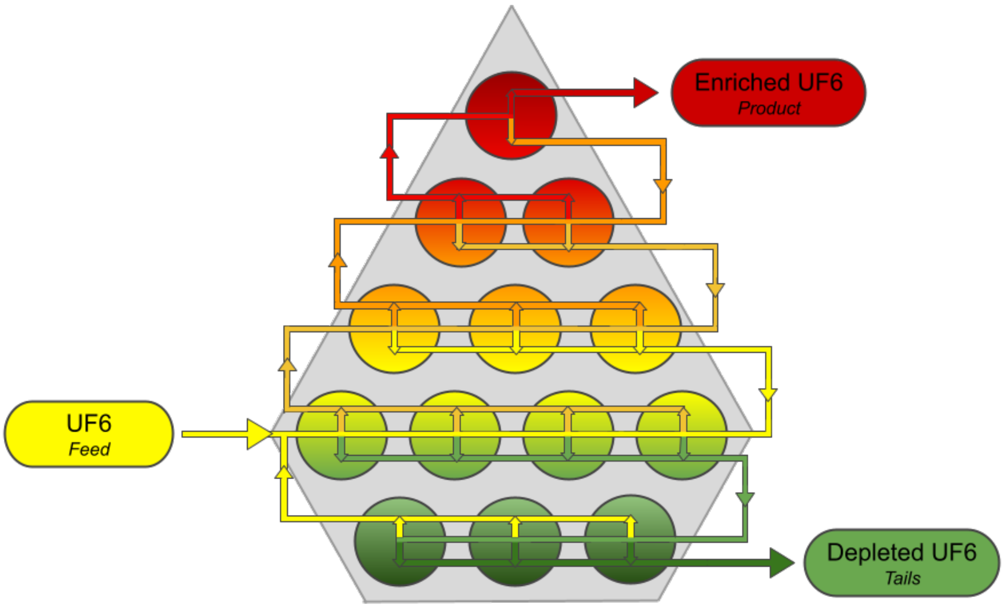


Figure 4: Example of a cascade for illustrative purposes.

1.2 Urenco

Urenco (**Ur**anium **en**richment **co**mpany) is an international supplier of enrichment services and fuel cycle products. Urenco was founded in 1970 after the governments of Germany, the Netherlands and the UK signed the Treaty of Almelo, which allows for international cooperation regarding centrifuge technology for enriching uranium [51].

Urenco has four enrichment facilities: Urenco Nederland (UNL) in Almelo, Urenco UK (UUK) in Capenhurst, Urenco Deutschland (UD) in Gronau, and Urenco USA in Eunice (UUSA). Each facility has multiple cascades used for uranium enrichment. The facilities UNL and UUK have been in operation since 1973. In addition to enriching uranium, UNL uses centrifuge technology to produce isotopes for medical, industrial and research applications. UD started production in 1983, and UUSA in 2010. The head office is located in Stoke Poges, UK [51].

Enrichment Technology Company (ETC) used to be a subsidiary of Urenco. It manufactures gas centrifuges. ETC was launched in 2003 and became a joint venture of Urenco and Orano (then still Avera) in 2006. ETC is responsible for installing the centrifuges in Urenco's cascades [11].

1.3 Motivation

Urenco experiences centrifuge crashes that not only affect the centrifuge itself but also damage surrounding centrifuges, which can undesirably shorten their lifespan. Although some centrifuges are preemptively stopped based on expert judgement, this approach is largely subjective and relies largely on individual experience rather than systematic assessment. Urenco is interested in developing a more proactive strategy that uses predictive techniques to anticipate centrifuge crashes before they occur.

1.4 Note from the Author

Non-proliferation is greatly concerned with the risk of centrifuge technology being used to produce high-enriched uranium for nuclear weapons. An international treaty aims to prevent the spread of nuclear weapons. Therefore, some details in this report are redacted and available in the confidential version. Such paragraphs, concepts, or variables are marked as *'confidential'*.

2 Problem Analysis

This section provides insight in the lifecycle of a centrifuge. It begins by outlining various operational scenarios (Section 2.1.1), the different maintenance actions (Section 2.1.2), types of failures (Section 2.1.3), root causes (Section 2.1.4) with the associated deterioration processes (Section 2.1.5). Additionally, the available data for machines is discussed (Section 2.1.6), possible approaches for data analysis (Section 2.2.2), the data preprocessing steps (Section 2.2.3), how we identify deviating centrifuges (Section 2.2.4) and a decomposition approach to analyse the data further (Section 2.2.5). The section closes by stating the research objective (Section 2.3), research questions (Section 2.4) and the structure of this report.

2.1 Life cycle of Centrifuges

2.1.1 Operational Situations

A centrifuge has different operational modes :

- **Run up**: The machine is turned on from a standstill and the rotor accelerates in several steps until the nominal frequency is reached. During a run up the machine passes critical phases that impose additional strain. Machines are run up after refurbishments.
- **Feed Inlet**: The machine is filled with UF6.
- **In operation**: The machine operates at nominal frequency. A machine is in operation during the enrichment process.
- **Run down**: The machine is powered off and the rotor decelerates to a complete stop. Run downs are executed for refurbishments or can occur unsolicited, classified as a failure.
- **Stand-by**: During refurbishments, the machine receives no power and stands still.

2.1.2 Maintenance Actions

A centrifuge cannot be repaired from failure. However, certain actions can be performed before failure:

1. *stop*: The operator turns the centrifuge off, causing it to run down. This can be done either to avoid failure or in preparation for a refurbishment.

Two other actions, *flitzebogen* and *ISCAR*, are possible. These are not included as the information to choose these actions is confidential. The above maintenance actions are done for each centrifuge individually. Maintenance actions can also be taken on cascade-level, such as refurbishments or connecting new feed material.

2.1.3 Failure Types

Each machine failure is classified by type. Two types of critical failures are distinguished based on rotor behaviour:

- CRASH: The rotor fails while working at a nominal frequency. The released energy causes a reaction where "crash gasses" are created and the rotor is destroyed.
- RUN DOWN: The centrifuge slows down due to the friction of the centrifuge exceeding the torque of the rotor. The rotor is nearly intact.

There exist hybrid failure types that are a combination of the above, but these can often only be identified by an autopsy, which takes a lot of work and time and is, hence, almost never done. Therefore, we neglect these hybrid types. A failure is documented as "Found not running", when it is unclear which failure type occurred.

2.1.4 Failure Causes

Different causes result in a machine failure. A failure cause is defined as the primary reason for the failure, although other factors may have contributed. The causes can be divided into two groups:

1. **Accident**: Sudden failures caused by nature or human events.
2. **Surface deterioration**: Failures that develop over time due to wear, friction, and lubrication.

Surface deterioration failures likely follow consistent patterns and are therefore predictable. Accidental failures are typically single events with no preceding evidence, making them harder to predict. Therefore, we focus on surface deterioration failures. We describe all potential failure causes and the related deterioration process for the surface deterioration group [1].

- **Manufacturing** or **Assembly Problems**: These defects are present from the beginning and don't develop over time. The failure occurs during the initial run up. This is categorized as an *accident* failure and can be a CRASH or RUN DOWN.
- **Feed Inlet/ Run Up**: A machine experiences more stress during a run up and run down as it passes some critical phases. Although caused by deterioration, the failure became critical due to a maintenance action, making the cause an *accident*. The rotor is still largely intact, so it is considered to be a RUN DOWN.
- **High Pressure**: A clogged scoop blocks material leaving the centrifuge, while the amount that enters remains the same. The holdup (the total amount of gas in the rotor) increases and so does the pressure. When the holdup exceeds the safety limit, the machine experiences a CRASH or RUN DOWN. The holdup develops over time, so the cause is due to *surface deterioration*.
- **Condensing Feed Impurities**: Pollution in the feed enters the rotor and builds up. An increasing CFI-indicator indicates this. The CFI-indicator can decrease, with the introduction of relatively clean feed. However, once the CFI-indicator exceeds a threshold, UF6 solidifies and the machine CRASHES. The pollution builds up over time, hence this cause is classified as *Surface deterioration*.
- **Light Gas**: Light gas is defined as any gas with a lower molecular weight than UF6 (e.g. oxygen, nitrogen, hydrogen fluoride) [34]. Light gas enters the centrifuge and accumulates, causing drag that eventually slows the rotor down. The result is a RUN DOWN. The buildup develops over time, so the cause is classified as *surface deterioration*.
- **Light Gas with a Pressure Pulse**: A crash creates light gases and a pressure pulse that can affect neighbouring machines. The result is a CRASH. The failure occurs due to the failure of another machine, so it is classified as an *accident*.
- **Low Temperature**: Recall that the material UF6 is in gaseous state. A too-low temperature in the centrifuge causes the UF6 to desublime in the rotor, leading to imbalance and a CRASH. The temperature changes over time, hence this cause is classified as *surface deterioration*.
- **High Temperature**: A higher temperature in a centrifuge has caused problems in other types of centrifuges. However, these issues have not been observed in TC-12 centrifuges. Therefore, we do not explore this cause further.
- **Frequency**: Before reaching the normal frequency, the rotor passes through multiple critical frequencies, which give extra stress to the rotor. If the rotor keeps operating at these frequencies, it would eventually lose balance and CRASH. In normal operation, machines do not operate at critical frequencies. Therefore, this cause is classified as an *accident*.

[1] A more elaborate description is given in the *confidential* report

- **Corrosion**: The rotor wall develops weak spots that eventually lead to holes. UF6 leaks from the rotor, causing friction and drag. Eventually, the motor RUNS DOWN when the drag exceeds the motor's torque. Corrosion develops over time, so this cause is a *surface deterioration*.
- **Moisture Ingress**: When refurbishments are not performed correctly, moisture enters the machine. UF6 reacts with moisture to form the solid Uranyl fluoride (UO2F2) and the highly corrosive hydrogen fluoride (HF) [34]. The machine would CRASH immediately. Since moisture is only present due to maintenance errors, the cause is classified as an *accident*.
- **Seismic Event**: Vibrations cause contact between rotating and stationary parts. The instability leads to a CRASH. Seismic events are caused by nature, so this cause is classified as an ACCIDENT.
- **Design Flaws**: Some machine versions have a reduced life span due to poor design. We will not consider centrifuges with design flaws and neglect this factor.

A summary of the deterioration causes along with the corresponding variables and effects is presented in Table 1.

Deterioration Cause	**Variable**	**Effect**	**Failure type**
High Pressure	Pressure	Holdup of UF6	CRASH or RUN DOWN
Condensing Feed Impurities	CFI-Indicator	Solid UF6 in rotor	CRASH
Light Gas	Amount of light gas	Drag in rotor	RUN DOWN
Low Temperature	Centrifuge inside temperature	Solid UF6 in rotor	CRASH
Corrosion	Rotor wall strength	Drag in rotor	RUN DOWN

Table 1: Deterioration causes with the corresponding effects.

The causes Frequency, High Temperature, Moisture Ingress, and Seismic Event have not occurred at UNL. These causes will therefore not be considered in the sequel.

2.1.5 Deterioration States

The surface deterioration causes from Section 2.1.4 are translated into individual paths with distinct deterioration states. Each path result in a CRASH (F_1) or RUN DOWN (F_2). The directed arcs in the graphs represent the possible transitions between states. Paths A, B, D involve a threshold that, once reached, leads to failure.

- **High Pressure**
  A high pressure can result in a CRASH or RUN DOWN. See Table 2 for the states and possible transitions in the path for High Pressure.

State	**Description**
A_0	Normal pressure level. During operation at nominal frequency, the pressure is *confidential* mbar.
A_1	Higher pressure results in more holdup (the amount of gas in the rotor). The motor consumes more power. The pressure is between *confidential* mbar and *confidential* mbar.
A_2	Pressure exceeds the threshold (*confidential* mbar). The holdup in the rotor is too much.
F1 or F2	Failure: CRASH or RUN DOWN

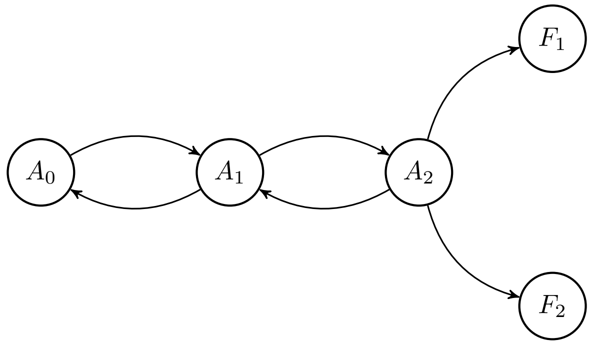


Table 2: High Pressure states and transitions.

- **Condensing Feed Impurities**
  Once a machine experiences pollution, there remains some degree of pollution and the CFI-indicator never returns to 0%. Therefore, the first state B_0 (see Table 3) combines zero increase with little increase.

State	Description
B_0	No (0%) or little increase ($\leq$ *confidential* %) in the CFI-indicator.
B_1	A high increase in CFI-indicator: between *confidential* % and *confidential* %. Machine experiences significantly more strain.
B_2	CFI-indicator exceeds the threshold (*confidential* %) and the UF6 solidifies.
F_1	Failure: CRASH

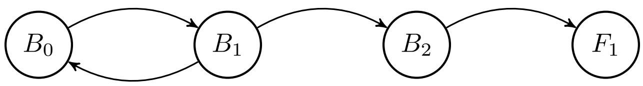


Table 3: Condensing Feed Impurities states and transitions.

- **Light Gas**
  Once a machine experiences light gas, there remains some degree of light gas. Therefore, the first state C_0 (see Table 4) combines no light gas and a small amount of light gas.

State	Description
C_0	No or a small amount of light gas. The rotor experiences little drag in the rotor.
C_1	A large amount of light gas that causes a lot of drag in the rotor.
C_2	The amount of light gas exceeds the threshold. The drag exceeds the torque of the motor and starts to run down.
F_2	Failure: RUN DOWN

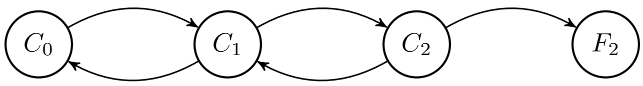


Table 4: Light Gas states and transitions.

- **Low Temperature**
  A small increase in temperature accelerates the corrosion process but has no known additional consequences. Therefore, a higher-than-usual temperature is considered to be within the normal temperature range in state D_0 (see Table 5).

State	Description
D_0	Normal temperature T in the rotor: *confidential*$°C \leq T$.
D_1	Lower temperature T . The machine still operates but colder material requires more power: *confidential*$°C < T \leq$ *confidential*$°C - 2°C$
D_2	The temperature T is below the threshold, and the UF6 desublimes: $T \leq$ *confidential*$°C - 2°C$.
F_1	Failure: CRASH

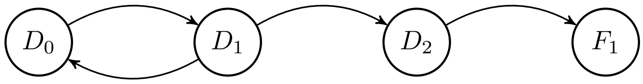


Table 5: Low Temperature states and transitions.

- **Corrosion**
  Corrosion is a monotone process, so once a state in path E is left, it cannot be returned to (see Table 6).

State	Description
E_0	No sign of corrosion.
E_1	Weak spots have developed in the rotor wall.
E_2	The weak spots have become holes. UF6 leaks through the rotor wall and causes drag.
E_3	The drag exceeds the torque of the motor and the run down starts.
F_2	Failure: RUN DOWN

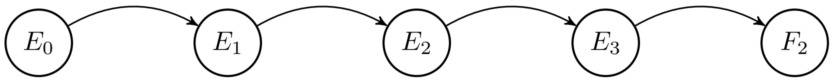


Table 6: Corrosion states and transitions.

2.1.6 Centrifuge Monitoring System

The main objective of the centrifuge monitoring system (CMS) is to inform operators about changes in the condition of centrifuges while centrifuges are in operation. The CMS measures various performance characteristics on centrifuge- and cascade level.

The **power** and **cos-phi** factor is measured for every centrifuge every 15 seconds. Data compression stores the current value every hour, along with any values significantly deviating from the last stored value. This method has been used for about a year. Previously, only one snapshot per day was stored. Operators compare the power with certain given boundary values to detect operational issues.

The cos-phi factor ($\cos\phi$) is the ratio of effective power P to apparent power S, see Figure 5. The effective power, (also known as active power) is the power actually used by the machine, while the apparent power S is the total power required. The cos-phi factor indicates how much power is lost during the transport of power. It is calculated as follows:

$$\cos\phi = \frac{P}{S}$$

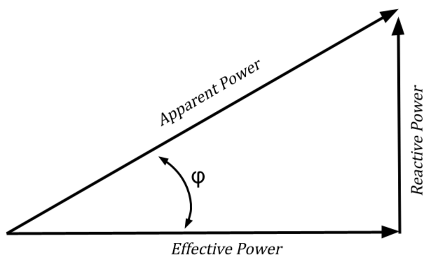


Figure 5: Power triangle.

The **frequency** of the rotor can be measured for each centrifuge, but not simultaneously with the power. Under optimal operation, the frequency remains constant. During run up and run down, the frequency increases and decreases, respectively.

The **CFI-indicator** is measured by the Cascade Contamination Monitoring System (CCMS) for a few centrifuges. These measurements are assumed to reflect the degree of pollution of machines with the same feed, which are the machines in the same flomel.

The outside **temperature** of some centrifuges is measured. Operators can also take manual temperature measurements using a thermographic camera, but these are not stored.

The **header pressure** of the whole cascade is measured. If it exceeds a threshold, a high-high alarm sounds and the cascade is evacuated.

The **feed cylinders** can be tested for material concentration to determine the amount of condensing feed impurities or light gas. However, the exact amount that enters a specific centrifuge remains unclear.

A **drag scan** measures the friction of a centrifuge. This action requires all the machines in a cascade to be drained of UF6, which leads to a production loss.

All actions taken on a machine or cascade are recorded in a logbook, including changes to operational settings and the introduction of new feed cylinders.

2.1.7 Pattern Profiling

We aim to determine whether the deterioration paths from Section 2.1.5 are observable by analyzing multiple run-to-failure time series and time series of machines that have not failed yet, but show deviating behaviour. Each failure is classified by type, but the specific cause of the CRASH or RUN DOWN is unknown.

Based on the deterioration paths from Section 2.1.5, we expect specific patterns to emerge as described in Table 7 [2]. Patterns are classified as long-term or short-term, with long-term covering multiple years, and short-term spanning several days or weeks. All centrifuges in a flomel receive the same feed, therefore differences between flomels is likely due to high pressure, light gas, or CFI. Differences within a flomel may be the result of corrosion or temperature.

[2] A more detailed description is given in the *confidential* report

Deterioration Cause	**Expected Pattern/ Characteristics**	Failure
Feed Inlet/Run up	The failure occurs after refurbishments	RUN DOWN/ CRASH
High Pressure	Short-term increase of the power consumption	RUN DOWN/ CRASH
Condensing Feed Impurities	The rotor experiences instability, so this may be reflected by random short-term fluctuations in power data	CRASH
Light Gas	The rotor experiences more drag, so the power short-term increases	RUN DOWN
Light Gas + Pressure Pulse	Failure must be preceded by another in the cascade. Higher power consumption due to the drag caused by light gas.	CRASH
Low Temperature	The power consumption increases as the temperature decreases. This can be a short-term or long-term increase.	CRASH
Corrosion	A long-term process, where the rotor eventually experiences drag due to leaks, leading to an increased power consumption.	RUN DOWN

Table 7: Expected patterns in power data per failure cause.

2.2 Data Analysis

2.2.1 Data

The data used for our data analysis are power time series from 15-second intervals and daily snapshots, both obtained from the centrifuge monitoring system.

The 15-second interval data contain power measurements recorded at 15-second intervals. However the data is compressed, so only the value for every hour is stored in the file, along with any values significantly deviating from the last stored value. Each centrifuge has one file, containing the entire period from March 2023 till May 2024.

Daily snapshot data contain daily power values. The values of all centrifuges in one cascade are stored in yearly files. The available data covers the period November 2008 to March 2023.

2.2.2 Pattern Detection Approaches

The goal is to establish a relation between the observations and the underlying deterioration state of a machine via data analysis. Several approaches exist [28]:

Visual representations of data are line graphs, bar charts and scatter plots. This is a simple approach to spot patterns and trends.

Statistical analysis can be divided in *descriptive statistics* and *inferential statistics.*

Descriptive statistics summarize numerical characteristics of a dataset for a basic understanding of the data's distribution. The summary often contains the mean, median and variability [28].

Inferential statistics uses techniques such as regression analysis, time series analysis, and hypothesis testing to identify correlations between variables. Regression analysis estimates the relation between a dependent and one or more independent variables. Time series analysis uses moving averages, exponential smoothing and linear regression to identify trend-, seasonal-, cyclical-, or irregular patterns. Hypothesis testing statistically evaluates assumptions about population parameters, such as whether the data follows a normal distribution [28].

Machine learning (ML) algorithms are useful for large and complex datasets. ML algorithms learn to make predictions or decisions based on past data. They are classified in *supervised-*, *unsupervised-* and *reinforcement learning* [28].

Supervised learning trains a model with labelled data. The algorithm learns to predict new, unknown, or unlabeled data. Supervised learning methods are regression, classification, decision trees, and deep learning.

Unsupervised learning trains a model with unlabelled data. The algorithm learns along-side unlabeled data to find patterns, similarities, or clusters. Unsupervised methods are clustering, principal component analysis, and kernel density estimation.

Reinforcement learning trains an agent in an environment. The algorithm learns and adapts its decision-making strategy using the feedback from the environment.

Data mining is used to identify correlations between variables that are unclear to the human eye. It is a sub-field of ML, but data mining is focused on uncovering patterns and insights. Examples are association rule mining and sequential pattern mining [28].

We expect deterioration patterns in the data, but it is unknown how these are portrayed. Therefore, any interesting variation, peaks, or sudden shifts are analyzed. We use visualisation to detect anomalies and trends. and *hypothesis tests* to quantify these observations and asses whether results are due to real effects or due to random variation. The null hypothesis (H0) states there is no significant difference or effect in the data. The alternative hypothesis (H1) claims that a substantial difference or effect does exist. The p-value quantifies the evidence against the null hypothesis. A lower p-value implies stronger evidence against H0, leading to a rejection when the p-value falls below a threshold α, known as the significance level, typically set at 0.01 or 0.05. The significance levels affects the occurrence of a type I error or type 2 error. A type I error is incorrectly rejecting H0, while a type II error fails to reject a false H0. The force of a statistical test depends on the sample size. Larger samples are more likely to detect true effects. The power also depends on the variability in the data samples. A higher variability can make it harder to detect an effect.

Tests can be either parametric, which often assume normally distributed data, or nonparametric, which make no assumptions about the underlying distribution. Common tests include the t-test for comparing means, chi-square test for independence, ANOVA for comparing multiple groups, and correlation tests for assessing relationships.

2.2.3 Preprocessing Data

The data should be cleaned and preprocessed to handle missing values, outliers, and obtain a format ready for analysis.

1. **Interpolation of missing values**: The data has two types of missing values. The first type consists of purposely missing values due to data compression. This results in a gap of at most an hour. Missing values have no significant deviations from the last known value. Therefore, Last Imputation Carried Forward (see Appendix D) can be used to impute the missing values. This method replaces missing values with the last known value. The second type of missing values are unsolicited missing values, often caused by system malfunctions. These data gaps that can span a day or multiple weeks [28]. There are multiple ways to handle missing values. If only a few consecutive values are missing, then an estimation can be calculated by interpolation, such as moving average or linear interpolation (see Appendix D for different imputation methods). When many consecutive values are missing, interpolation methods may be inaccurate. The variable(s) can be dropped from the analysis, but if all machines have the same gap we have no data left. Therefore, we should be aware of the gap and take this deficiency into consideration.

2. **Eliminate outliers**: The power operates within specific limits. Any values measured outside this range are considered as outliers and removed.

3. **Rescale data**: In this report version, data values are scaled to a value between 0 and 1, allowing for unrestricted analysis and discussion of the results. This is done by multiplying each value in the dataset by a constant f.

4. **Eliminate noise**: The power depends on different factors specific to a machine. Further details can be found in the *confidential* report. Some power patterns result from different operational settings or actions, such as introducing new feed material. These patterns are considered noise for the data analysis. Such operational actions are recorded in a logbook.

   One of these actions is measuring the frequency of the machines. This interrupts the motor drive, and causes the motor to run asynchronous. To return to the synchronous state, the motor requires relatively more power, which is visible by peaks as shown in Figure 6. The peaks inflicted by operators are removed, while asynchronous peaks that appear in a single time-series are likely due to deterioration and are retained.

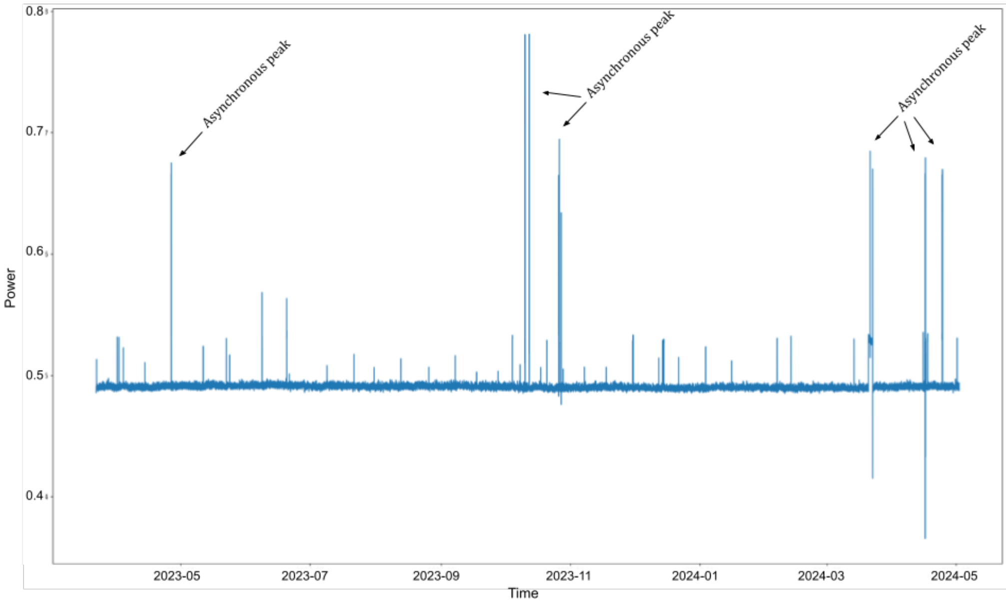


Figure 6: Power of a centrifuge, scaled with factor f.

2.2.4 Identifying Deviating Centrifuges from Daily Snapshot Time Series

Given the large amount of available data, we focus on one cascade and limit to centrifuges that have failed or show deviating behaviour.

Centrifuges with deviating behaviour are identified by analysing daily snapshot data and comparing centrifuges within a flomel and comparing flomels. In Figure 7, are all 16 centrifuges shown for four flomels. We see differences between flomels, e.q. the all time series in the bottom left-flomel have a monotone decreasing trend, while time series from the other flomels swing from decreasing to increasing multiple times. The three red-outlined centrifuges show from a certain year an increasing trend. One of these deviating centrifuges is shown in Figure 8, where the long term increasing and decreasing parts are indicated by, repectively the green and red subseries. A long-term increasing trend may indicate corrosion (as we expected in Section 2.1.7). Deviating centrifuges are further analysed on the 15-second data.

2.2.5 Decomposition of 15-Second Time Series

We aim to identify the states from Section 2.1.5 in the time series to calculate the corresponding transition probabilities. We believe state transitions are in some way anomalies in the time series. We analyse individual time series by answering the following questions:

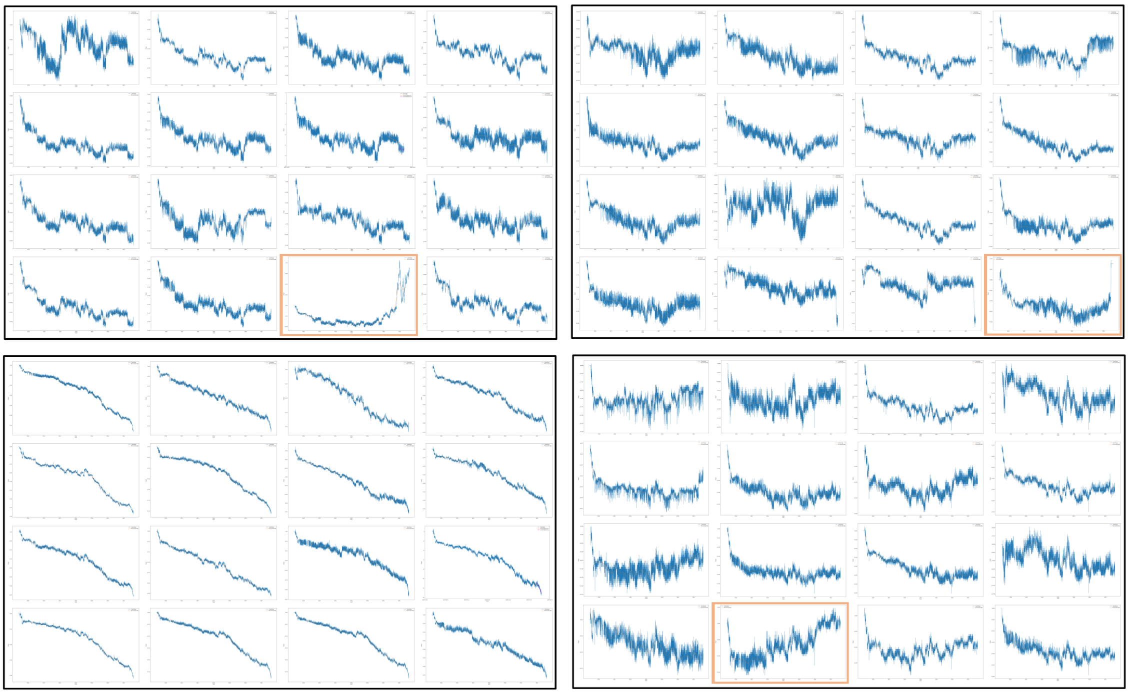


Figure 7: Daily snapshots are plotted over the period from 2008 to 2023 for 4 flomels (each with 16 centrifuges) . Three time series that show deviating behaviour are red-circled time series. The top left flomel is from the depleting stages, The bottom left is from the feed stage, and the right flomels are from the enriching stage.

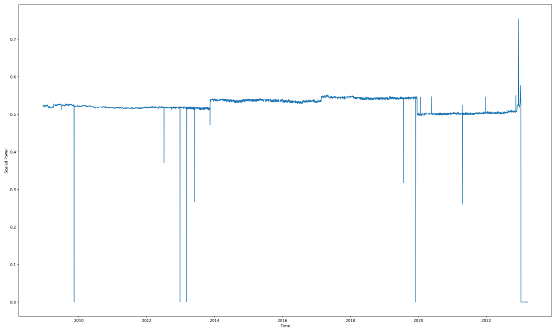
1. *Does the frequency of asynchronous peaks increase?*
2. *Does a trend exists?*
3. *Does a seasonal pattern exists?*
4. *If so, is the seasonal pattern constant over time?*
5. *Does an asynchronous peak affect the seasonal pattern?*
6. *Does a medium peak affect the cyclical pattern?*
7. *Are anomalies observable in the residuals?*

For Question 1, we classify a data point as asynchronous when it deviates a certain factor from the long term trend. The factor is determined via the density plot. In the density plot the time series looks like a normal distribution with a long tail that contains a bump that can be assigned to the asynchronous peaks.

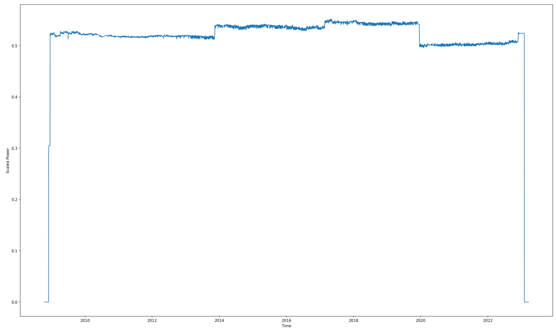
We determine whether there is a trend in the frequencies and duration of these peaks by testing these on stationarity using the Augmented Dickey-Fuller test and a KPSS test, which are explained later.

Questions 2-7 are answered by decomposing the time series, which is suitable for identifying underlying relations. Decomposition methods are based on the assumption that observed data contain four components:

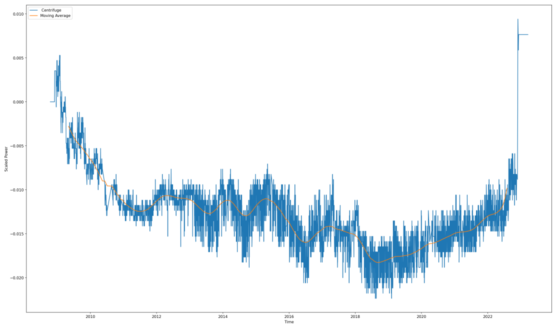
1. Trend T_t: Long-term change in the time series.
2. Cycle C_t: Fluctuations with a non-fixed frequency. The cyclic component is often included in the trend component.
3. Seasonality S_t: Repeating pattern with a fixed and known frequency. A seasonal component is generally shorter than the cyclic component



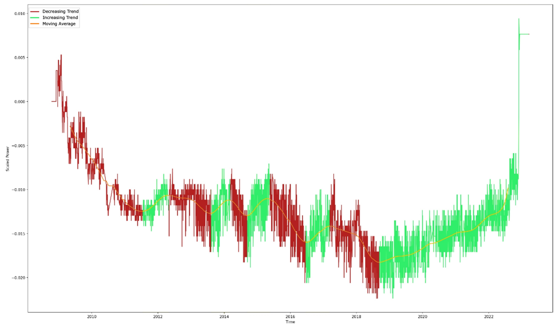
(a) Raw daily snapshot data



(b) Daily snapshot without outliers



(c) Daily snapshot without steps



(d) Increase and decrease in time series trend.

Figure 8: Daily snapshot time series plotted from 2008 to 2023.

4. Residuals e_t: The remainder between the above components and the observed data.

The functional relationship between these components can have different forms. An additive or multiplicative model is often distinguished, but combinations of additive and multiplicative models also exist. An *additive model* assumes the above components work independently and is useful when the seasonal variation is constant over time. [33]:

$$X_t = T_t + S_t + C_t + e_t, \quad t = 1, 2, \ldots$$

A *multiplicative model* assumes dependent components, which is often the case for real-world problems. The trend is non-linear and seasonality varies in frequency and/or amplitude:

$$X_t = T_t \times S_t \times C_t + e_t, \quad t = 1, 2, \ldots$$

To determine whether a multiplicative or additive model is more appropriate, a Shapiro or Jarque-Bera test is performed on the values and log values (depending on the length of the time series). If the values are sampled from a normal distribution, an additive model is appropriate. While if the log values are sampled from a normal distribution, a multiplicative is more appropriate.

The **Shapiro-Wilk Test** is a normality test appropriate for small sample sizes [6].

H0: Sample is derived from a normally distributed population.
H1: Sample is not derived from a normally distributed population.

The **Jarque-Bera Test** is a goodness-of-fit test that determines whether the sample data is normally distributed by checking the skewness and the kurtosis. It is suitable for large sample sizes with over 2000 samples for which other normality tests like Shapiro-Wilk are unreliable [6].

H0: Skewness is zero and excess kurtosis is zero, hence sample is normally distributed
H1: Sample is not derived from a normally distributed population

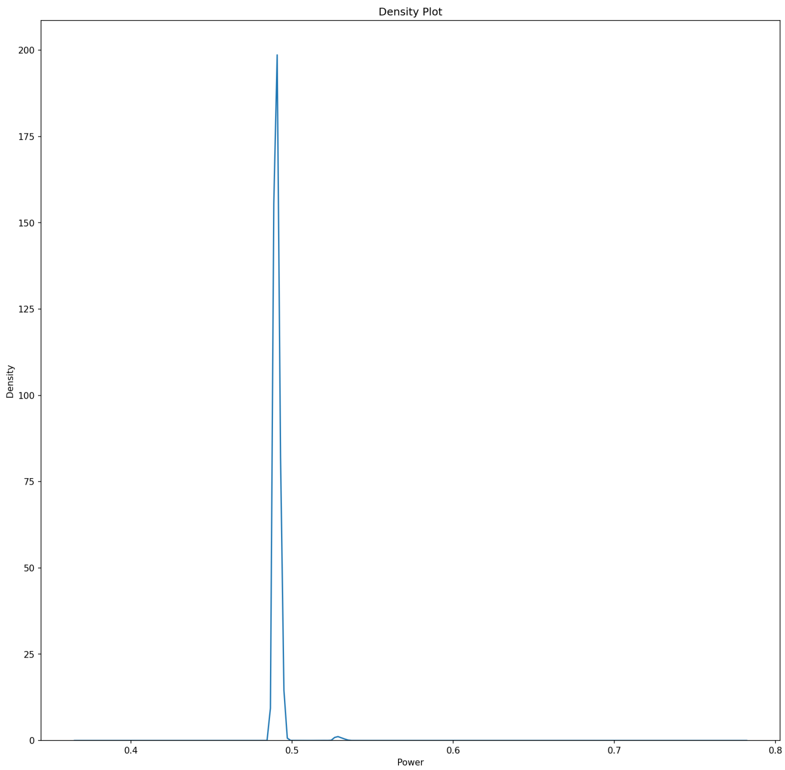


Figure 9: Density plot of the scaled power.

If the hypothesis is rejected for the values and the log values, then the choice of a multiplicative or additive model is based on the variance of the time series. An increasing variance implies that the four components are not constant over time and a multiplicative model is more appropriate. For a constant variance an additive model is suitable. The variance is tested using Levene's test.

Levene's test assess the variances of two or more groups on equality:

H0: All samples are from a population with equal variances

H1: The variances of the samples are not equal

Now we can decompose the time series using the classical decomposition approach by Brockwell and Davis in [6], while also including statistical tests to substantiate evidence. A flow diagram of the analysis is shown in Figure 10.

Step 1: Trend T_t

The Augmented Dickey-Fuller test and Kwiatkowski-Phillips-Schmidt-Shin test are used to ensure the series is truly stationary. The **Augmented Dickey-Fuller test** (ADF) is used to determine whether the time series is non-stationary, i.e. a trend exists. The ADF test uses an autoregressive model.

H0: Time series has a unit root

H1: Time series has no unit root

If the time series has a unit root, it impliesthe time series is non-stationary. For a long time series the ADF test will most likely find evidence for some non-stationary, therefore we apply the test on daily snapshots.

The **Kwiatkowski-Phillips-Schmidt-Shin** (KPSS) test

H0: Time series is stationary

H1: Time series is non-stationary

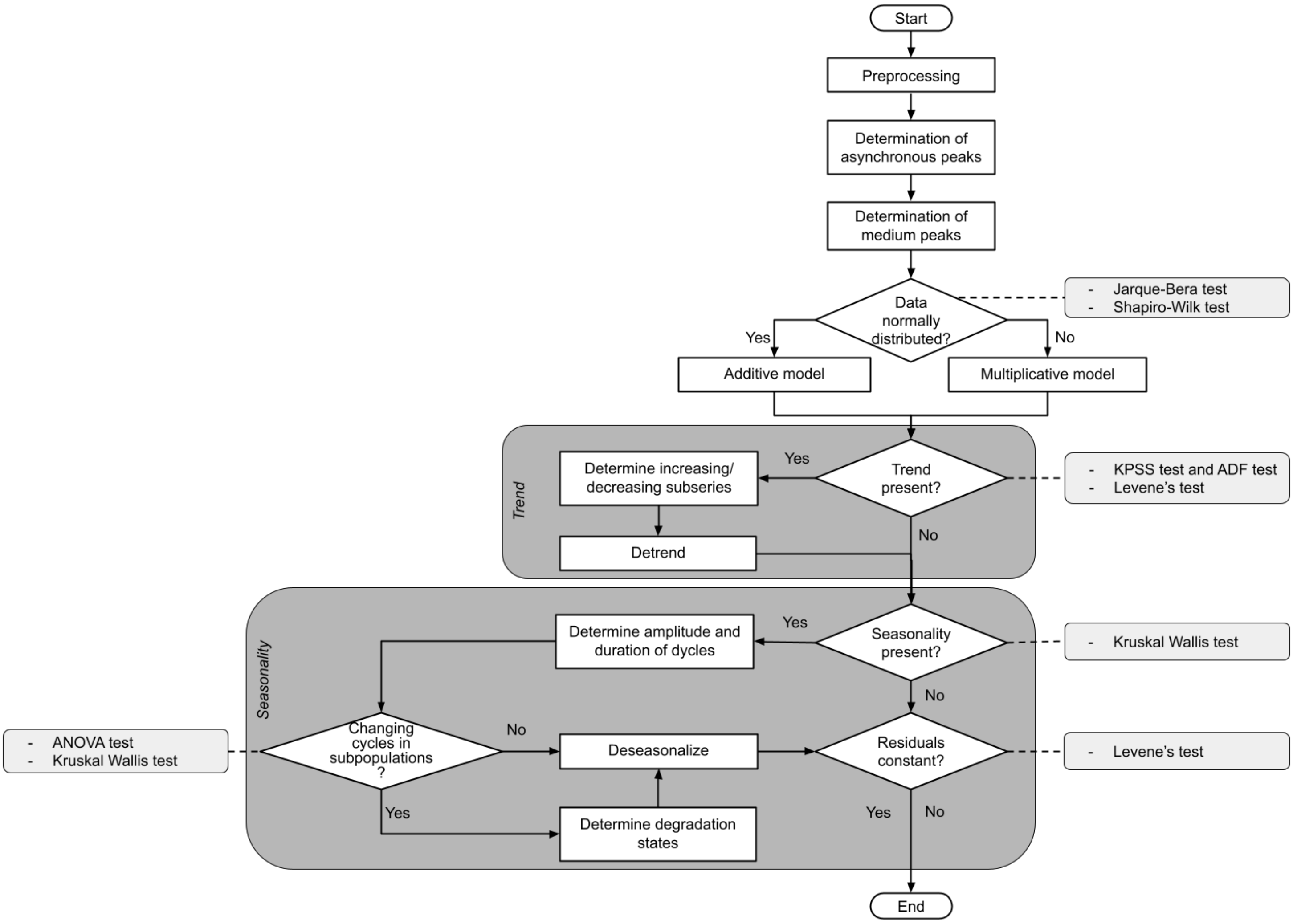


Figure 10: Flow diagram of the 15-second decomposition method.

If both tests conclude the series is (not) stationary, then the series is (not) stationary. If KPSS indicates stationary and ADF not, then the series is trend stationary and the trend needs to be removed for strict stationary. If ADF indicates stationary and KPSS not, then the series is difference stationary, and the series is differenced.

When we have established that a trend exists, we estimate the trend T_t using a moving average filter with window size d. That is,

$$\hat{T}_t = \begin{cases} \frac{1}{d}\left(\frac{1}{2}X_{t-q} + X_{t-q+1} + \ldots + \frac{1}{2}X_{t+q}\right) & \text{for} \quad d = 2q, \quad q < t < n-q, \\ \frac{1}{d}\left(X_{t-q} + X_{t-q+1} + \ldots + X_{t+q}\right) & \text{for} \quad d = 2q+1, \quad q+1 < t < n-q, \end{cases}$$

The window size d determines the smoothness of the curve. It should be of the same size or a multiple of the seasonal length to not include the seasonal component in the trend component. The observations are detrended: $X_t - \hat{T}_t$ for an additive model, $X_t/\hat{T}_t$ for a multiplicative model with $t = 1, 2, \ldots$.

The trend T_t is again smoothed with a moving average window to determine increasing and decreasing subsequences.

Step 2: Seasonality S_t

Determining whether seasonality exists without visualisation can be done by the **Kruskal-**

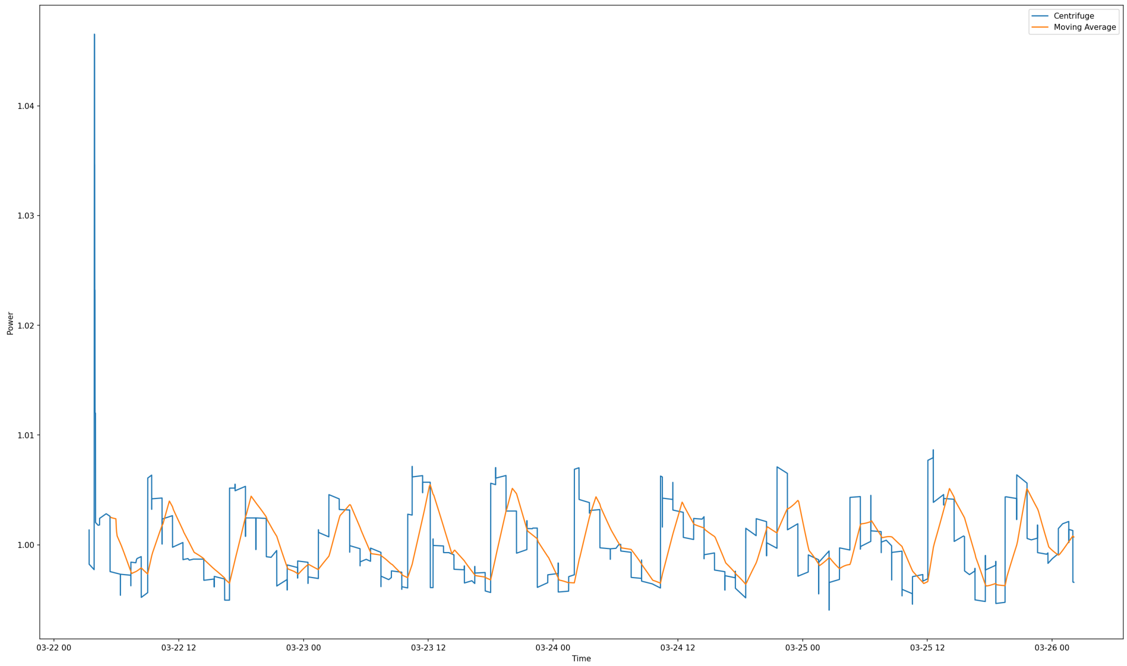


Figure 11: The blue plot is the detrended power time series, and the orange plot is the moving average of the time series. Seasonal cycles are visible. At the start is a medium peak.

Wallis test. This is a non-parametric statistical test. It assumes a stationary time series.

H0: Medians between groups are equal

H1: Medians between groups are not equal

To determine whether the pattern is seasonal and not cyclical, we perform Levene's test on the frequencies of each cycle. The variances of the amplitude is also tested.

We noticed medium peaks in our time series. The parts of the time series between the medium peaks are considered as subpopulations. The subpopulations are tested on a trend in durations or amplitude values using the ANOVA and Kruskal Wallis test

Estimate seasonal effects S_k for $k = 1, \dots, d$. Fast Fourier Transform (FFT) can be used to identify the most critical frequency in periodic signals. The averages of the detrended series $(X_l - \hat{m}_l)$ are estimated and used to determine the seasonal component S_k. Let $q < l = k + jd \leq n - q$, $j = 0, 1, \dots$ then

$$\hat{S}_k = \begin{cases} \overline{(X_l - \hat{m}_l)_k} - \frac{1}{d}\sum_{i=1}^{d} \overline{(X_l - \hat{m}_l)}_i, & k = 1, \dots, d, \\ \hat{S}_{k-d} & k > d. \end{cases}$$

The observations are also deseasonalized for an additive model: $D_t = (X_t - \hat{m}_t) - \hat{S}_t, \quad t = 1, \dots, n$. And for a multiplicative model by $D_t = X_t / \hat{S}_t, \quad t = 1, \dots, n$

Step 3: Residuals e_t

Now that the time series is detrended and deseasonalized, the residuals e_t remain. A changing variance in the residuals implies that the seasonal component is changing. This is tested with Levene's test.

The time series in between the medium peaks are considered as subseries.

A **one-way ANOVA test** (**AN**alysis **O**f **VA**riance) determines whether there exists statistically significant difference between mean values of three or more groups It is an extension of the t- and z- test methods. It assumes the data is normally distributed.

H0: Means of all populations are equal $\mu_1 = \mu_2 = \ldots = \mu_k$

H1: At least one population mean is different from the rest

Revisiting Questions 1-7 listed at the beginning of this subsection, we find that

1. *Does the occurrence of asynchronous peaks increase?*
   After all non-centrifuge-specific asynchronous peaks are removed, often only a pair of asynchronous peaks remain, so that there is not enough evidence to make conclusions about the occurrence of these peaks.
2. *Does a trend exists?*
   Most centrifuge time series are non-stationary and therefore have a trend. This is not a strictly monotone trend over the complete time period. We encountered a handful of time series that were considered stationary after preprocessing.
3. *Does a seasonal pattern exists?*
   All analysed centrifuges returned enough evidence for a seasonal component according to the Kruskal Wallis test. From each seasonal pattern the amplitudes and seasonal durations are retrieved by finding all peaks and determining the height and the time interval between these peaks.
4. *If so, is the seasonal pattern constant over time?*
   The mean durations and amplitudes of the seasonal cycles for each subseries are non-stationarity. So the seasonal patterns change over time.
5. *Does an asynchronous peak affect the seasonal pattern?* Similar to Question 1, after all non-centrifuge-specific asynchronous peaks are removed, often only a pair of asynchronous peaks remain, so that there is not enough evidence to make conclusions about the effect of the asynchronous peaks.
6. *Does a medium peak affect the seasonal pattern?* Some populations had enough statistical evidence for a trend in the duration or amplitude of the seasonal component. Some subpopulations showed non-stationary behaviour, these are used to retract states.
7. *Are anomalies observable in the residuals?*
   No anomalies are visually observable. The variance is not constant within the time series, which again proves that the seasonal component is changing.

2.2.6 State Determination

We assume based on the pattern profiling from Section 2.1.7, the relations between the time series and the underlying state are as described in Table 8. The time series denote from which type of data we retrieve the states.

Path	Deterioration Cause	Time series	Relation to time series
A	High Pressure	15-second	Increasing amplitude of the cycles from the seasonal component
B	Condensing Feed Impurities	15-second	Increasing/decreasing duration of cycles from the seasonal component
C	Light Gas	15-second	Short-term increasing trend
D	Low Temperature	15-second	Long short-term increasing trend
E	Corrosion	Daily snapshots	Long-term increasing trend

Table 8: Relation between the paths and the time series.

The transitions for each individual path are retrieved using the decomposition method. So for each centrifuge we retrieve the transitions in dimension A, B, C,D, and E. Then the transitions are merged together for the multi-dimensional state (A, B, C, D, E).

2.2.7 Kaplan-Meier Estimator for Survival Times

Survival analysis assesses the effect of an intervention by measuring the number of subjects surviving over some time. For survival, the variable of interest is the time that elapses before some event occurs. We define an event as a transition in the machine's deterioration state. The analyses are complicated when not all machines have experienced an event or failure before the end of the study. These cases are right-censored observations for which partial information is available. It is undesirable to exclude these subjects because they provide some information about survival and the sample size may become too small. Further, the hazard rate would be overestimated if the empirical average is computed.

There is data available for the whole lifespan of a machine, but not every type of data. The install date is known for every centrifuge and for some cascades, the daily snapshots are stored since the day operation started. The 15-second data is only available since March 2023. Therefore, it depends on the type of statistical tests and analysis whether the data is censored. The failure pattern of centrifuges is known to follow a bath tub curve (Appendix F). The beginning of its lifetime is the 'Infant Mortality Stage' which mostly contains failures due to defective components. This is followed by the 'Random Failure Period' with a low failure rate. The 'Wear-out Stage' is the period of our interest as it contains mostly deterioration failures. Therefore, the beginning of a lifetime is not as relevant to our problem as the end.

Censored data is either left-, right, or interval censored (see Figure 12).

- Left censored data is data for subjects that started before the start of the observation.
- Right censored data is data for subjects that have not failed yet at the end of the observations, so the failure time is unknown.
- Interval censored data is data where the exact failure time is unknown, but the lower- and upper bounds of an interval surrounding the failure are known.

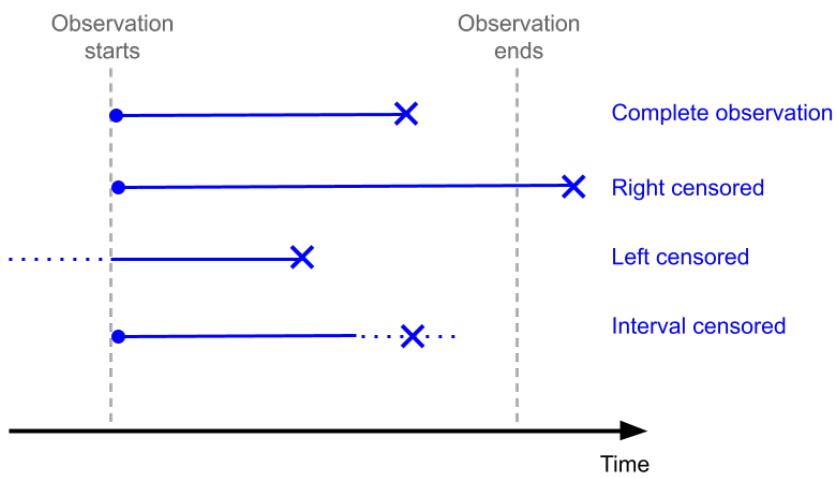


Figure 12: Examples of censored lifetimes. The blue dot is the start of the lifetime, and the cross represents the occurrence of an event. The solid blue lines are known periods of time, while the dashed blue lines represents an unknown period.

The Kaplan-Meier survival curve computes the survival over time despite the above difficulties. The Kaplan-Meier survival curve is the probability of surviving in a given length of time while considering time in many small intervals. It is assumed that censored subjects follow the same survival prospects as completely observed subjects. The survival probabilities are equal for subjects that start at different times. We assume events occur at the end of the interval (the first observable time).

The survival probability at a time interval t is calculated by the number of surviving subjects divided by the total number of subbjects at the start:

$$S_t = \frac{\text{\# of subjects operating at the start} - \text{\# of failed subjects at time } t}{\text{\# of subjects operating at the start}}$$

Subjects that have died, before the start are censored and not counted in the denominator. The total probability of survival till a time interval is the multiplication of all probabilities of survival at all time intervals preceding that time [19].

2.3 Research Objectives

The objective is to prevent centrifuge crashes by proactively stopping centrifuges to avoid damage to neighbouring centrifuges. We aim to determine a predictive maintenance policy that provides the optimal stopping time for a centrifuge given its current state of deterioration. The policy must balance the trade-off between the risk of failure and extended operating hours. Stopping a centrifuge too early results in unnecessary production losses, while stopping it too late increases the risk of a crash.

2.4 Research Questions

The main research question is:

“ Can we determine a policy to prevent centrifuge crashes based on real-time observations? ”

The associated sub-questions are:

1. What kind of model can be used to determine a predictive maintenance policy?
2. How can a centrifuge’s remaining useful life distribution be described?
3. What structure has the optimal policy for a single centrifuge?
4. How can find the optimal policy?
5. Does the policy structure change when a system of centrifuges is considered?

2.5 Structure of the report

The remainder of this thesis is structured as follows:

Section 3 reviews literature and approaches in the domain of maintenance optimization.

Section 4 examines deterioration models for maintenance optimization and examines preliminary concepts and notions of Markov decision theory, including Markov chains.

Section 5 describes a model for the single-unit and multi-unit problem, and proves the structure of the corresponding optimal policy.

Section 6 provides the results for the models introduced in Section 5.

Section 7 summarizes the key results and provides suggestions for further research.

Section 8 summarizes the research and answers the problem statements from Section 2.

3 Related Works

3.1 Maintenance Models

Developments in technical systems and increasing reliance on equipment requires effective planning of maintenance activities. De Jonge et al. [13] review maintenance optimization models classifying them into single- and multi-unit systems. Further sub-classification is based on the structure of the deterioration state space, which can be discrete or continuous.

Discrete state spaces are classified in two, three, or more states. A two-state model usually has a functioning and failed state. Three state deterioration processes have a state between the functioning and failed state. For example, Zhang et al. [55] find a preventive maintenance policy for a three-state single-unit system. The deterioration process is separated in a continuous deterioration process caused by wear-and-tear and a discrete deterioration caused by shocks, which is modelled by a Wiener and compound Poisson process. The resulting policy triggers repair or replacement actions based on age and condition level, where the condition level is monitored via discrete inspections. In continuous state spaces the deterioration level can take any value within a certain range. Single-unit systems with a discrete state space are mainly approached using Markov decision processes, where actions are based on the condition of the unit. The number of deterioration states is related to what can be measured in practice. Continuous state spaces are often discretized or some approximation technique is used to solve the problem [15, 13].

Maintenance actions are *preventive* or *corrective*. Preventive actions are performed before failure of a unit and are triggered based on time, usage, or condition information. For example, Cha et al. [8] consider age-based maintenance for a single-unit system with an increasing failure rate, where external shocks, modelled by a Poisson process, lead to an increase in the failure rate. Corrective maintenance is performed after failure and performed on units that do not deteriorate, such as electronics [2].

A unit is *repairable* or *non-repairable*. Non-repairable units can only be maintained through replacement. A repair maintenance action can be *perfect* or *imperfect*. A perfect repair returns the state to as-good-as-new. While an imperfect repair improves the state but not to the as-good-as-new condition. For example, Zhou et al. [57] consider a multi-unit system that deteriorates according to a continuous-time Markov chain and determine a preventive maintenance threshold for imperfect maintenance using linear programming. Finkelstein [16] compares perfect and imperfect repair based on the cost function. He describes how the nature of the problem differs, as a perfect repair corresponds to a renewal process, while an imperfect repair leads to an nonhomogeneous Poisson process. Finkelstein also discusses on the existence of an optimal degree of imperfect repair for a system.

In reality, units rarely operate in isolation. Maintenance models often distinguish three types of dependencies: *structural*, *stochastic*, and *economic*. Multiple dependencies may exist in a single problem, but most literature only assumes one type to reduce complexity. Structural dependence describes the difference in operation between machines and is often considered in multi-unit systems. For instance, van Staden et al. [52] address structural dependence by dividing the machines in groups based on technology type and power output level. Stochastic dependence exists when the deterioration or failure of multiple units are dependent. This type of dependence is rarely considered as it quickly leads to complex structures [13]. Economic dependence is often considered in the cost function of the model and reflects the economic interactions between units regarding maintenance. For example, Lugtigheid et al. [31] consider a fixed maintenance set-up cost that is 'shared' when multiple units are maintained.

Various studies describe the deterioration process with a discrete-time Markov chain (DTMC). For example, Jimenez et al. [25] use a DTMC to describe the deterioration of sewer pipes.

Proportional hazard models also appear regularly to describe the deterioration process. Here, the failure rate depends on a unit's age and condition that is modeled by a Markov chain. These models assume that the total failure rate of a unit is a baseline failure rate multiplied with a functional term that models the systems characteristics. Vlok et al. [53] use a Weibull proportional hazard model to determine the optimal replacement policy for a unit based on monitored vibrations levels.

Semi-Markov decision processes extend these models by considering state transitions that are not limited to fixed time intervals. This allows for a more realistic modeling of a unit's deterioration. For example, Chen and Kishor [9] consider a single-unit system with random decision epochs, where minimal maintenance or major maintenance can be performed. They find that the optimal policy follows a threshold-type policy.

Makis and Jiang [32] consider imperfect inspections using a Partially Observable Markov Decision Process (POMDP), where the observed state is stochastically related with the actual state, accounting for uncertainties in condition monitoring. Kim [27] uses a POMDP for imperfect condition signals and approaches this using Bayesian learning.

3.2 Optimal Stopping

Optimal stopping problems and optimal replacement problems play an important role in predictive maintenance models, as they aim to determine the best time to take an action. The timing of actions can greatly impact the overall efficiency and costs.

Optimal replacement problems are a specific type of optimal stopping problems where the only maintenance action is a replacement of a component. For example, de Saporta describes [14] an optimal replacement problem for a degrading component using a three-dimensional Piecewise Deterministic Markov Process. The model balances costly early interventions and too late interventions causing system failures.

Aven [3] describes an optimal replacement policy for a general failure model with two states. A component degrades according to a stochastic process that is interrupted by an observable stochastic underlying condition, The failure rate and costs depend on the system's condition. Aven also considers a system with multiple failure modes, and a case where replacements may not occur immediately.

3.3 Structure of Optimal Stopping Policies

An MDP allows consideration of the trade-off between immediate and expected future costs when choosing an action. The optimality equations can be used to establish that the objective function has certain properties as a function of state and action, e.g. subadditivity. These properties can guarantee that the optimal policy has a certain simple form, such as a threshold policy.

Oh and Özer [38] study discrete-time optimal stopping problems and propose a method to characterize the structure of the optimal stopping policies with a finite horizon. A set of meta-theorems establish conditions on the state transition and the one step value function. The method is demonstrated on various stopping problems, such as the option pricing problem, the dynamic market entry model, and the secretary problem An outline is given to extend the results to infinite horizon problems.

Hjort et al. [24] provide conditions for monotone policy for a system with a totally ordered multi-state space and independent components. Lindqvist [30] generalize this and consider a partially ordered state spaces. The result is shown for a repairable systems with dependent components. The system process is modeled by a Markov chain. Both discrete and continuous time problems are considered and conditions are imposed on the transition matrix.

In discrete-time problems, often the history dependence causes a multidimensional state. Christensen and Irle [10] study certain multidimensional optimal stopping problems in discrete and continuous time using the monotone case approach. This approach uses Doob decomposition of the reward function and is applied to versions of the house-selling and burglar's problem. Christensen and Irle note that optimal stopping theory reaches its limits for finding explicit solutions for multidimensional problems.

4 Theoretical Background

We outline approaches for maintenance optimization problems (Section 4.1), including deterioration models for failure prediction (Sections 4.2- 4.5). Next, we describe a Markov decision process (Section 4.6) and an optimal stopping problem (Section 4.7) with the various solution methods for determining an optimal policy (Section 4.8).

4.1 Maintenance optimization

Maintenance models determine when to inspect, repair, and replace deteriorating units. Maintenance optimization aims to improve these policies. Sirakar et al. [49] define three maintenance strategies;

- Corrective Maintenance (CM): Maintenance is performed after a unit fails;
- Preventive Maintenance (PM): Maintenance is performed before failure occurs. The strategies are time- or usage-based; e.g. "replace a unit after X hours" or "replace a unit after Y operation cycles";
- Predictive Maintenance (PdM): The unit's condition is measured and its reliability is assessed. Maintenance actions are scheduled based on predicted future conditions. This includes Reliability Centered Maintenance (RCM) and Condition-Based Maintenance (CBM).

We aim to prevent crashes by making decisions based on current observations, using a PdM strategy. "Prediction" involves using analytical, statistical and machine-learning models for *diagnosis* and *prognosis*. Diagnosis identifies the unit's condition, while prognostics estimate the time to failure. Together, they allow for remaining useful life estimations [56].

PdM techniques include knowledge-based, model-based, physical-based, data-driven, and hybrid. These approaches are described in Table 9.

Technique	**Model**	**Data requirements**	**Knowledge requirements**
Knowledge-based	Expert-system	No data	Domain knowledge
Model-based	Statistical distribution based models	Lifetime data and dependencies between units	Understanding of the deterioration process
Physical-based	Models based on laws of physics	Observations of physical wear-and-tear patterns	Domain knowledge in physics and deterioration
Data-driven	Machine-Learning model for failure pattern recognition	Large amount of run-to-failure histories	-
Hybrid	Combination of multiple techniques	See the requirements of the considered techniques	See the requirements of the considered techniques

Table 9: Predictive Maintenance techniques with the corresponding requirements [29].

4.2 Remaining Useful Life

Remaining useful life (RUL) is the time a unit is expected to function before failing. RUL prediction models are key for anticipating maintenance needs in PdM strategies [4].

RUL is estimated from observations, average estimates of similar units, or a combination of both. The accuracy depends on factors, such as the machine type, data quality, and modelling technique. Two types of techniques are often distinguished [17]:

- **Model-based** techniques use a deterioration model aligned with the physical structure of the system.
- **Data-driven** techniques assume no knowledge about the wear-and-tear process and predicts the RUL based only on past observations.

For complex systems, it is difficult to accurately model the deterioration process. Data-driven methods overcome this difficulty and are suitable when the deterioration process is too complex or not understood. However, data-driven methods often ignore uncertainties in material properties, measurement errors, operating conditions, and obtaining sufficient data for rare failure causes can be difficult [17, 23].

The choice of method also depends on the type of available data, which can be categorized into three types [4]:

- **Lifetime data** from similar machines contains the length of the operational period from start-up to failure. The data is suitable for **Survival Models**, a statistical approach that requires few data sets. Examples include Proportional hazard models and probability distributions like Weibull or exponential;
- **Run-to-failure histories** from similar machines are used in **Similarity Models**. These data-driven models capture deterioration patterns and match new data with these patterns to find the closest profile.
- **Prescribed threshold value data** contains information on critical conditions. For example, "the machine must not exceed 160°F to avoid failure". Condition indicators are extracted from sensor data and used to fit **Deterioration Models**, which predict when the threshold is exceeded. Thresholds can also be calculated, given there is enough statistical data.

Survival Models fit a model or probability distribution on the data. But in practice, different runs can vary greatly in duration or shape. Similarity models handle this by matching deterioration patterns. However, similarity models are data-driven and function as a black box, lacking insight into the system's internal structure [50]. We present model-driven approaches to describe a deterioration process of a machine.

4.3 Deterioration Models

Deterioration models describe and predict the physical conditions of equipment. It supports decision makers to understand how fast a condition aggravates or violates a threshold. These models are influenced by *sampling* and *temporal uncertainty* [41]. Sampling uncertainty refers to the variability in deterioration between samples. This epistemic uncertainty can be reduced by increasing the amount of samples. Temporal uncertainty refers to the uncertain progression of deterioration over time. This is a random type of uncertainty and cannot be completely reduced by considering more samples.

A unit's condition can be represented by a deterministic index or a failure probability. Traditional deterioration methods are usually deterministic, while more recent methods use probabilistic models.

- **Deterministic models** are deterioration curves based solely on factors like age. These are known as failure patterns. No probabilities can be incorporated and the uncertainty about variables is ignored. Examples are the bathtub curve and P-F curve. See Appendix F for more details on deterministic models.
- **Probabilistic models** describe the probability of being in a certain condition. The probability distributions can be updated as new information becomes available. This ensures the distributions represent the system's current state at any time. Probabilistic models are categorized into *random variable* (RV) and *stochastic process models.*

RV models randomize parameters of an empirical deterioration law to reflect the variability among deterioration samples. It cannot consider temporal uncertainty, so a specific sample path is considered deterministic.

Stochastic process models consider temporal uncertainty, keeping specific sample paths uncertain. Examples are Markov chains and Gamma processes.

It is important to consider the uncertainties related to deterioration for meaningful reliability analysis. If temporal uncertainty is present, a stochastic process is most suitable [41].

4.4 Stochastic Process Models

A stochastic process evolves over time based on probabilistic rules, making it partially random.

Definition 4.1. A stochastic process is a family of random variables $\{X_\theta\}$ with $\theta \in \Theta$. For discrete-time processes, Θ consists of integers. For a continuous time process, Θ is a real line and then θ is often replaced by t: $\{X(t)\}$ [48].

For stationary stochastic processes, the probabilistic rules do not change over time. These models can be discrete or continuous in time and state space. A discrete-state, discrete-time model allows for direct numerical calculations and simulations. Often it is also easier to determine transition probabilities for discrete-states than transition densities for continuous state spaces. Future states of stochastic processes are either dependent or independent of past and present states.

4.5 Markov Chain

A Markov chain is a type of Markov process, which is a stochastic processes with the Markov property.

Definition 4.2. The *Markov property* states that future behaviours of the system only depends on the current state of the model, and not its history behaviour.

A Markov chain models systems transitioning among a finite number of states. Each transition is called a *step* and the system can be in one state at a time [54]:

Definition 4.3. A discrete-time stochastic process X_t is a *Markov chain* if

$$\mathrm{P}\left(X_{n+1} = i_{n+1} | X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n\right) = \mathrm{P}\left(X_{n+1} = i_{n+1} | X_n = i_n\right),$$

with $i_0 \leq i_1, \ldots, i_{n+1} \in S$, here S contains all possible states of the process.

One way to represent the transition probabilities is in a transition matrix, also known as a probability- or Markov matrix [20].

Definition 4.4. The *transition matrix* P for the Markov chain is the $N \times N$ matrix, whose (i, j)-th entry $\mathrm{P}_{ij}$is the transition probability for moving from state i to state j, and $0 \leq \mathrm{P}_{ij} \leq 1$, for $1 \leq i, j \leq N$ and $\sum_{j=1}^{N} \mathrm{P}_{ij} = 1$ for $1 \leq i \leq N$.

The transition matrix contains one-step probabilities. To find the probability of transitioning from state i to state j in n periods is given by

$$\mathrm{P}_{ij}^{(n)} = \sum_{k \in S} \mathrm{P}_{ik}^{(n-1)} \mathrm{P}_{kj},$$

or by matrix multiplication

$$\mathbf{P}^{(n)} = \mathbf{P}^{(n-1)} \mathbf{P}^{(1)}.$$

Definition 4.5. Let X be a Markov chain with state space S. If there exists some $m \geq 0$ for some states $i, j \in S$, such that $\mathrm{P}_{ij}^{(m)} > 0$, then i leads to j. If j also leads to i, then i and j are said to *communicate* with each other.

States that can be transitioned into and out of are called *transient*. States that cannot be transitioned out of are *absorbing* states. A special type of Markov chain is an absorbing Markov chain. This is a Markov chain with at least one absorbing state and from every state it is possible to transition to an absorbing state in several steps. In Figure 13 an example of an absorbing Markov chain is visualized with one absorbing and $\Delta - 1$ transient states.

Suppose an absorbing Markov chain has r absorbing states and t transient states, then the transition matrix, where the first t stats are transient and the last r absorbing, has the following form [20]:

$$\mathbf{P} = \begin{pmatrix} \mathbf{Q} & \mathbf{R} \\ \mathbf{0} & I \end{pmatrix}, \tag{1}$$

with $\mathbf{Q}$ an t-by-t matrix, $\mathbf{R}$ a nonzero t-by-r matrix, $\mathbf{0}$ the r-by-t zero matrix, and $\mathbf{I}$ is the r-by-t identity matrix.

The n-step transition matrix has the form

$$\mathbf{P}^n = \begin{pmatrix} \mathbf{Q}^n & \sum_{i=0}^{n-1} \mathbf{R}\mathbf{Q}^i \\ \mathbf{0} & I \end{pmatrix}.$$

The probability of being in a transient state approaches zero as the number of steps approaches infinity.

Theorem 4.1. *In an absorbing Markov chain, the probability that the process will be absorbed is 1 [20]:*

$$\mathbf{Q}^n \to \mathbf{0} \quad as \quad n \to \infty.$$

The proof can be found in Appendix G

The following theorems can be used to determine the expected number of steps before absorption.

Theorem 4.2. *In an absorbing Markov chain, the matrix* $\mathbf{I} - \mathbf{Q}$*, where* $\mathbf{Q}$ *is defined in Equation* (1)*, has an inverse known as the fundamental matrix* $\mathbf{N}$*. Matrix*

$$\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-1} = (\mathbf{I} + \mathbf{Q} + \mathbf{Q}^2 + \ldots),$$

where the (i, j)*-th entry gives the expected number of times the process is in transient state* s_j*, given the process started in transient state* s_i*. The initial state is also counted in the case* $i = j$ *[20].*

The proof can be found in Appendix G.

Theorem 4.3. *Let* t_i *be the expected number of steps before the chain is absorbed starting from state* s_i*. The column vector* $\mathbf{t}$ *whose* i*-th entry is* t_i*. Then* $\mathbf{t} = \mathbf{N}\mathbf{e}$*, with* $\mathbf{N}$ *being the fundamental matrix for* $\mathbf{P}$ *and* $\mathbf{e}$ *being the all-ones column vector [20].*

Theorem 4.4. *Let* $\mathbf{B}$ *be an* $t \times r$ *matrix, with entries* $b_{i,j}$ *that denothe the probability that an absorbing chain is absorbed in absorbing state* s_j*, given starting in transient state* s_i*. Then*

$$\mathbf{B} = \mathbf{N}\mathbf{R},$$

where $\mathbf{N}$ *is the fundamental matrix and* $\mathbf{R}$ *as in canonical form [20].*

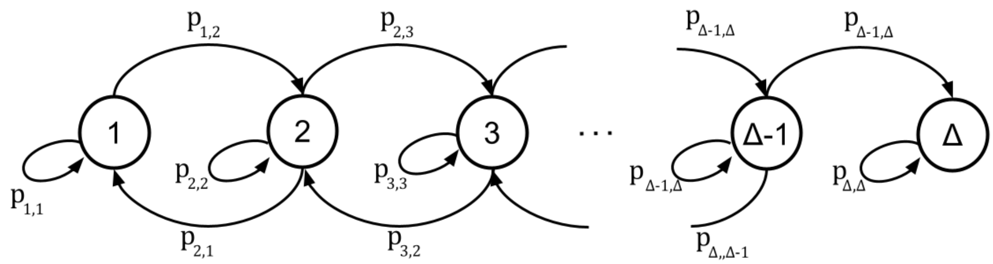


Figure 13: Absorbing Markov chain of a non-monotone deterioration process. The circles represent the possible states, the directed arcs visualise the possible transitions between the states, with $p_{i,j}$ being the probability of transitioning from state i to state j.

4.6 Markov Decision Process

A Markov decision process (MDP) models sequential decision making in systems with uncertain future states. At decision epochs, the operator observes a state, performs an action and the system transitions to the next state with a corresponding reward [5].

An MDP is a tuple $(\mathcal{S}, \mathcal{A}, p, r)$, where

- The planning horizon T is the set of all decision epochs. The set may be finite or infinite, and discrete or a continuum. If $T = \{1, 2, \ldots, N\}$ for some $N < \infty$, the problem has a *finite horizon*. No decision has to be taken at the last decision epoch N. If the set with decision epochs is infinite $T = \{1, 2, \ldots\}$, the problem has an *infinite horizon*. Discrete decision epochs require a decision at every epoch. While in a continuum, decisions are to be made either at all decision epochs, random points of times when an event occurs, or times chosen by the operator.
- $\mathcal{S}$ is the *state space*. This set contains all possible states. The state captures the configuration of the system.
- $\mathcal{A}$ is the *action space*. This set contains all possible actions. The set $\mathcal{A}_s$ contains all actions available while being in state s. Actions may be chosen randomly or deterministically.
- $p_t(s'|s, a)$ is the probability at epoch t for transitioning to state s', given current state s and choosing action a. It is assumed that $\sum_{s' \in \mathcal{S}} p_t(s'|s, a) = 1$.
- $r_t(s, a)$ is the reward at epoch t for being in state s and taking action a. The reward may depend on the state of the system at the next decision epoch. The expected value at decision epoch t can be estimated by

$$r_t(s, a) = \sum_{s' \in \mathcal{S}} r_t(s, a, s') p_t(s'|s, a).$$

  Instead of maximizing a reward function, one can minimize a cost function $c_t(s, a)$, which is the negative of the reward function.

The operator starts in an initial state $s_0 \in \mathcal{S}$. At each decision epoch t, an action $a_t \in A(s_t)$ is chosen. The system transitions to the next state s_{t+1} according to the transition function. An immediate reward r_t is received.

The decision rule $d_t : \mathcal{S} \to \mathcal{A}_s$ is a function that maps each state $s \in \mathcal{S}$ to an action $a \in \mathcal{A}_s$ at decision epoch t. Decision rules range from deterministic Markovian to randomized history dependent. This depends on how past information is incorporated and how actions are selected:

- Deterministic Markovian (MD) rules depend on previous system states and action only by the current state s_t and an action a_t is chosen with certainty.

- Deterministic history dependent (HD) rules depend on the history h_t of the system as represented by the sequence of previous states and actions, so $h_t = (s_1, a_1, \ldots, s_{t-1}, a_{t-1}, s_t)$, with s_i and a_i being the state and action at decision epoch i, respectively.
- Randomized Markovian (MR) rules are similar to MD rules, except actions are chosen according to a probability distribution based on the state s_t.
- Randomized history dependent (HR) is similar to HD, except actions are chosen according to a probability distribution based on history.

Any optimal problem can be made Markov by including all relevant information from the past in the current state [46].

A policy π is a sequence of decision rules for every decision epoch $\pi = (d_1, d_2, \ldots, d_{N-1})$, $N \leq \infty$. A policy is *stationary* if the decision rules are independent of time, and hence the same for every decision epoch: $\pi = (d, d, \ldots)$. The goal is to find a policy that optimizes an objective function over the time horizon [46].

The standard form of an optimization problem is

$$\begin{aligned} \text{minimize} \quad & f(\overrightarrow{x}) \\ \text{subject to} \quad & \overrightarrow{x} \in \Omega. \end{aligned}$$

The function $f : \mathbb{R}^n \to \mathbb{R}$ is the *objective function* and the vector $\overrightarrow{x} \in \mathbb{R}^n$ contains the decision variables. The optimization problem aims to find the best $\overrightarrow{x}$ in all possible vectors Ω. The objective function is also referred to as the value function.

The objective function can be described by maximizing the discounted expected rewards (or minimizing expected costs):

$$v_\gamma^\pi(s) = \lim_{N\to\infty} \mathbb{E}_s^\pi \sum_{t=1}^{N} \gamma^{t-1} r(s_t, \pi(s_t))$$

The discount factor $\gamma \in [0, 1)$ is introduced to limit the contribution of future rewards $r(s_t, \pi(s_t))$.

The objective function can also maximize the average expected reward:

$$v_\gamma^\pi(s) = \lim_{N\to\infty} \frac{1}{N} \mathbb{E}_s^\pi \sum_{t=1}^{N} r(s_t, \pi(s_t))$$

This approach is suitable for frequent decision making or when the discount factor is close to 1.

The Bellman equations, also known as the optimality equations, splits the value function into immediate rewards and discounted rewards for future states. The discounted optimality equations is the following system of equations [46]:

$$v_\gamma(s) = \max_{a\in\mathcal{A}_s} \{r(s, a) + \sum_{s'\in\mathcal{S}} \gamma p(s'|s, a) v(s')\}, \quad s \in \mathcal{S}. \tag{2}$$

For the average reward case, define $g = \lim_{\gamma\uparrow 1} (1-\lambda) v_\gamma^*(0)$ and $h(s) = \lim_{\gamma\uparrow 1} [v_\gamma^*(s) - v_\gamma^*(0)]$. Given these limits exist, the average reward optimality equations are

$$h(s) = \max_{a\in\mathcal{A}_s} \{r(s, a) - g + \sum_{s'\in\mathcal{S}} p(s'|s, a) h(s')\}, \quad s \in \mathcal{S}. \tag{3}$$

4.7 Optimal Stopping Problem

An optimal stopping problem is a Markov Decision Process with a finite state space and only two actions for each state: *stop* or *continue*. Such a problem is sometimes referred to as a *binary* *decision process*. The system evolves according to an uncontrolled process with a possibly non-stationary Markov chain. The goal is to find the optimal time to stop a given Markov process that minimizes a certain cost associated with the process. The action *stop* corresponds to a cost, while the action *continue* corresponds with a reward. The problem can have a finite or infinite horizon depending on the class of stopping times. When the stopping action is chosen, the process stops and no more costs or rewards are incurred [15, 46].

The optimal policy is often a monotone policy and since we have only two actions (a_0 and a_1), the monotone policy is a threshold policy. Here, the state space is divided into two mutually exclusive subspaces based on a threshold, where each action is only taken in one of the subspaces:

$$\pi(s) = \begin{cases} a_0, & s < \tau, \\ a_1, & s \geq \tau. \end{cases}$$

Finding an optimal policy reduces to finding a threshold τ, also called a *control limit*.

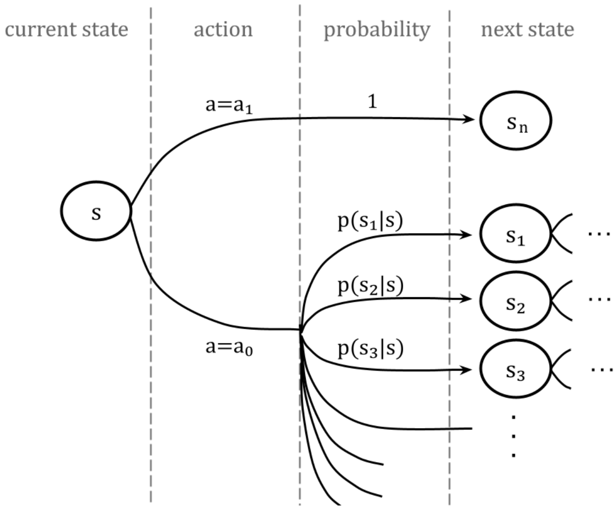


Figure 14: Symbolic representation of the decision process for optimal stopping. Here $s, s_1, s_2, \ldots, s_n \in \mathcal{S}$, where s_n is the absorbing state.

4.8 Policy making

Three main approaches exist to find optimal policies for finite MDPs: value iteration, policy iteration, and linear programming. Value iteration and policy iteration are both dynamic programming algorithms.

Value iteration finds the optimal policy by constantly solving optimality equations for each state. The actions with the most optimal value are determined and used in further calculations. The resulting policy is ϵ-optimal [46].

Policy iteration starts with a certain initial policy. The policy's value is fixed and used to determine a new policy. So it finds the optimal policy by iteratively computing v^n by $(I - \gamma P_{d_n})v = r_{d_n}$ and d_{n+1} by $d_{n+1} \in \arg\max_{d \in \mathcal{A}_s}(r_d + \gamma P_d v^n)$, until the optimal policy is found. This approach applies to stationary infinite-horizon problems.

Linear Programming finds the minimum or maximum value of an objective function subject to some constraints. The primal problem is often difficult to compute, therefore the dual program is solved instead. The result is an optimal policy.

5 Methodology

We begin by describing a Markov Decision Process (MDP) for a single centrifuge (Section 5.1, proving the structure of its optimal policy (Section 5.2) and describing the policy iteration algorithm to find the optimal solution (Section 5.3). Next, we extend this to an MDP for a system of centrifuges (Section 5.4), the optimal policy's structure is proven (Section 5.5), and the adapted policy iteration algorithm is described (Section 5.6). *Note*: A centrifuge is now referred to as a unit and a cascade is referred to as a system.

5.1 MDP: Single-Unit System

Our goal is to determine the optimal time to stop a unit to prevent a crash. We model this problem using a Markov Decision Process (MDP), where an operating unit carries a risk of crashing, while a non-operating unit produces no output. The state of each unit is described using the five deterioration paths outlined in Section 2.1.5.

5.1.1 Decision Epochs

The system is observed every 15 seconds, so we consider discrete decision epochs. The stochastic process ends when a unit fails or the stopping action is chosen. Units have a finite lifetime, but the exact duration is uncertain. Therefore, we assume an infinite set of discrete decision epochs $T = \{1, 2, \ldots\}$.

5.1.2 State Variables

The state variable $s \in \mathcal{S}$ describes the unit's condition. Two different states are directly observable: *Fault* and *Operating*. One operating state does not accurately describe the unit's health. Therefore, multiple operating states are defined that correspond with the deterioration processes in Section 2.1.4. We consider five deterioration paths: A: High Pressure, B: Condensing Feed Impurities, C: Light gas, D:Low Temperature, and E: Corrosion. These paths may evolve simultaneously, so the state is five-dimensional. As soon as one dimension has reached the failure state, the unit completely fails. So two failure states are defined: F_1 for a CRASH, and F_2 for a RUN DOWN. The state space is

$$\mathcal{S} = \left\{ \left\{ (A_i, B_j, C_k, D_l, E_m) \quad \text{for} \quad i, j, k, l \in \{0, 1, 2\}, \quad m \in \{0, 1, 2, 3\} \right\} \cup \{F_1, F_2\} \right\}.$$

5.1.3 Decision Variables

Two types of actions can be taken in all the states except the failure states:

- a_0: *continue*. No action is taken, there is no effect on the current state.
- a_1: *stop*. The unit is stopped preemptive via a Run Down.The next state is $s' = F_2$ with probability 1.

Once the absorbing state F_1 or F_2 is reached, the stochastic process stops and no more decisions have to be taken. So

$$\mathcal{A}_s = \mathcal{A} = \{a_0, a_1\}, \quad s \in \mathcal{S} \setminus \{F_1, F_2\}.$$

5.1.4 Transition Function

Observations are made every 15 seconds. Given this timescale, it is unlikely that multiple states in one dimension are crossed at once. However, for Condensing Feed Impurities and Low Temperature, solidification occur within seconds once the critical state is reached. So states B_1 and D_1 can transition to F_1 in one time step without observing the intermediate step. We also assume transitions occur in at most one dimension at a time.

The state variable is updated with the transition probabilities:

$$p(s'|s,a) = \begin{cases} p(s'|s), & s' \in \mathcal{S}, \quad a = a_0, \quad s \in \mathcal{S} \setminus \{F_1, F_2\}, \\ 1, & s' = F_2, \quad a = a_1, \quad s \in \mathcal{S} \setminus \{F_1, F_2\}, \\ 0, & \text{otherwise.} \end{cases}$$

The transition function $p(s'|s)$ is a multidimensional Markov chain. Let $s = (A_i, B_j, C_k, D_l, E_m)$ with $i, j, k, l \in \{0, 1, 2\}$, and $m \in \{0, 1, 2, 3\}$ (unless otherwise specified). Further, $p+q+r+s+t = 1$ with $p, q, r, s, t \in \{0, 1\}$, then all possible transitions are:

$$p(s'|s) = \begin{cases} p(s|s), & \\ p((A_{i+p}, B_{j+q}, C_{k+r}, D_{l+s}, E_{m+t})|s), & i+p, j+q, k+r, l+s \in \{0,1,2\}, \quad m+t \in \{0,1,2,3\}, \\ p((A_{i+p}, B_{j+q}, C_{k+r}, D_{1-s}, E_{m+t})|s), & i+p, j+q, k+r \in \{0,1,2\}, \quad m+t \in \{0,1,2,3\}, \\ p((A_{i+p}, B_{j+q}, C_{k-r}, D_{l+s}, E_{m+t})|s), & i+p, j+q, k-r, l+s \in \{0,1,2\}, \quad m+t \in \{0,1,2,3\}, \\ p((A_{i+p}, B_{j+q}, C_{k-r}, D_{1-s}, E_{m+t})|s), & i+p, j+q, k-r \in \{0,1,2\}, \quad m+t \in \{0,1,2,3\}, \\ p((A_{i+p}, B_{1-q}, C_{k+r}, D_{1-s}, E_{m+t})|s), & i+p, k+r \in \{0,1,2\}, \quad m+t \in \{0,1,2,3\}, \\ p((A_{i+p}, B_{1-q}, C_{k+r}, D_{1-s}, E_{m+t})|s), & i+p, k+r \in \{0,1,2\}, \quad m+t \in \{0,1,2,3\}, \\ p((A_{i+p}, B_{1-q}, C_{k-r}, D_{l+s}, E_{m+t})|s), & i+p, k-r, l+s \in \{0,1,2\}, \quad m+t \in \{0,1,2,3\}, \\ p((A_{i+p}, B_{1-q}, C_{k-r}, D_{1-s}, E_{m+t})|s), & i+p, k-r \in \{0,1,2\}, \quad m+t \in \{0,1,2,3\}, \\ p((A_{i-p}, B_{j+q}, C_{k+r}, D_{l+s}, E_{m+t})|s), & i-p, j+q, k+r, l+s \in \{0,1,2\}, \quad m+t \in \{0,1,2,3\}, \\ p((A_{i-p}, B_{j+q}, C_{k+r}, D_{1-s}, E_{m+t})|s), & i-p, j+q, k+r \in \{0,1,2\}, \quad m+t \in \{0,1,2,3\}, \\ p((A_{i-p}, B_{j+q}, C_{k-r}, D_{l+s}, E_{m+t})|s), & i-p, j+q, k-r, l+s \in \{0,1,2\}, \quad m+t \in \{0,1,2,3\}, \\ p((A_{i-p}, B_{j+q}, C_{k-r}, D_{1-s}, E_{m+t})|s), & i-p, j+q, k-r \in \{0,1,2\}, \quad m, m+t \in \{0,1,2,3\}, \\ p((A_{i-p}, B_{1-q}, C_{k+r}, D_{l+s}, E_{m+t})|s), & i-p, k+r, l+s \in \{0,1,2\}, \quad m+t \in \{0,1,2,3\}, \\ p((A_{i-p}, B_{1-q}, C_{k+r}, D_{l+s}, E_{m+t})|s) & i-p, j-q, k+r, l+s \in \{0,1,2\}, \quad m+t \in \{0,1,2,3\}, \\ p((A_{i-p}, B_{1-q}, C_{k-r}, D_{l+s}, E_{m+t})|s), & i-p, j-q, k-r \in \{0,1,2\}, \quad m+t \in \{0,1,2,3\}, \\ p((A_{i-p}, B_{1-q}, C_{k-r}, D_{1-s}, E_{m+t})|s), & i-p, j-q, k-r \in \{0,1,2\}, \quad m+t \in \{0,1,2,3\}, \\ p(F_1|s), & i = 2, \quad j,k,l \in \{0,1,2\}, \quad m \in \{0,1,2,3\} \\ & \vee \quad j \in \{1,2\}, \quad i,k,l \in \{0,1,2\}, \quad m \in \{0,1,2,3\} \\ & \vee \quad l \in \{1,2\}, \quad i,j,k \in \{0,1,2\}, \quad m \in \{0,1,2,3\}, \\ p(F_2|s), & i = 2 \quad j,k,l \in \{0,1,2\}, \quad m \in \{0,1,2,3\} \\ & \vee \quad k = 2 \quad i,j,l \in \{0,1,2\}, \quad m \in \{0,1,2,3\} \\ & \vee \quad m = 3, \quad i,j,k,l \in \{0,1,2\}, \\ 0, & \text{otherwise.} \end{cases}$$

Example

Suppose we only consider deterioration paths A and E. The state space is

$$\mathcal{S}_A \times \mathcal{S}_E = \Big\{ \Big\{ (A_i, E_m) \quad \text{for} \quad i \in \{0,1,2\} \quad \text{and} \quad m \in \{0,1,2,3\} \Big\} \cup \{F_1, F_2\} \Big\}.$$

This contains 14 states, two of which are failure states. Figure 15 shows the corresponding two-dimensional Markov chain for $a = a_0$. The possible transitions for 15-second-intervals are represented by the directed arcs.

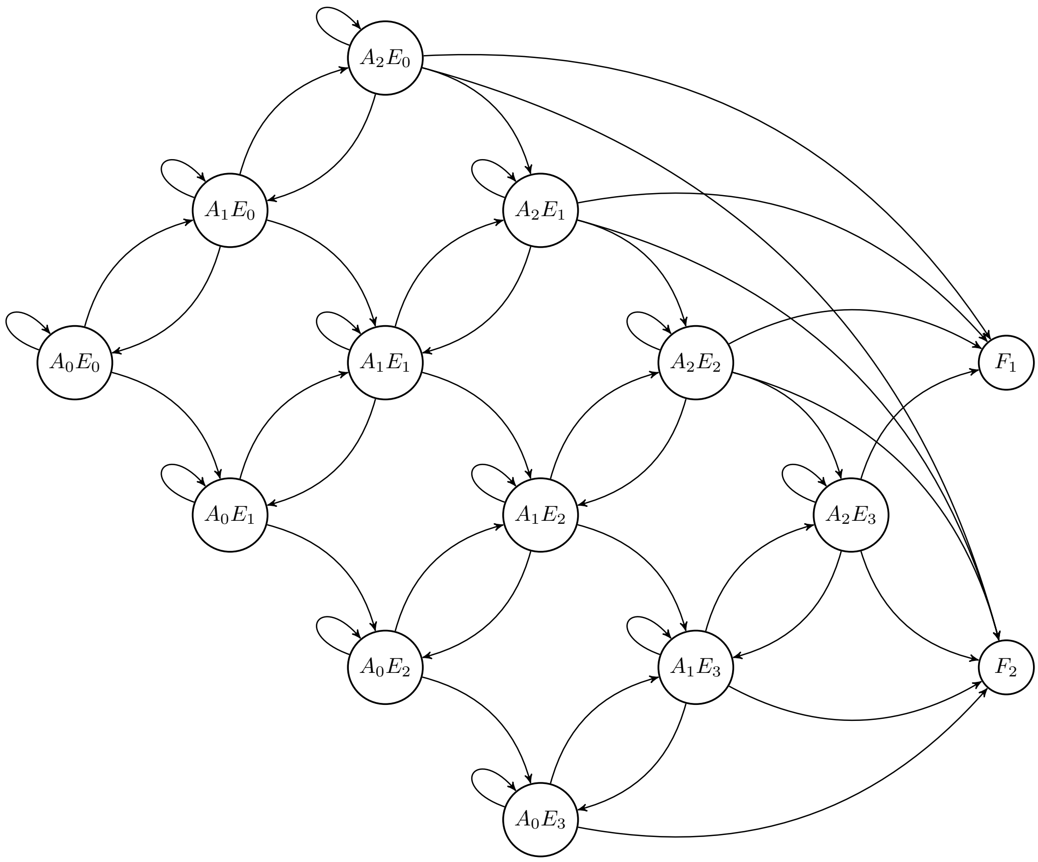


Figure 15: The circles represent the possible states, the directed arcs visualise the possible transitions between the states when action a_0 is taken.

5.1.5 Cost Function

The cost function considers rewards for production, and a penalty for unexpected crashes, which depends on the next state. Further, the production and therefore rewards nonincrease with the partial ordering of the unit's state. For current state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$, the cost function $c(s, a)$ is defined as

$$\begin{aligned} c(s,a) &= c_{\text{product}}(s,a) + c_{\text{crash}}(s,a) \\ &= c_{\text{product}}(s,a) + \sum_{s' \in \mathcal{S}} c_{\text{crash}}(s,a,s')p(s'|s,a), \end{aligned} \tag{4}$$

with

$$c_{\text{product}}(s,a) = \begin{cases} -c_1(s), & s \in \mathcal{S} \setminus \{F_1, F_2\}, \quad a = a_0, \\ 0, & \text{otherwise.} \end{cases}$$

$$c_{\text{crash}}(s,a,s') = \begin{cases} C_3, & s' = F_1, \quad a = a_0, s \in \mathcal{S} \setminus \{F_1, F_2\}, \\ 0, & \text{otherwise.} \end{cases}$$

Then Equation (4) reduces to:

$$c(s,a) = \begin{cases} -c_1(s) + C_3 \;\; p(F_1|s), & s \in \mathcal{S} \setminus \{F_1, F_2\}, \quad a = a_0, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The penalty C_3 for a CRASH is higher than the product reward $c_1(s)$ and all costs are finite. The production reward are defined as negative costs. This is to ensure that there is an optimum stopping interval for minimum costs. So $0 \leq c_1(s) \leq C_3 < \infty$.

5.1.6 Value Function

The value function for policy π is the γ-discounted expected costs. For $s \in \mathcal{S}$,

$$\begin{aligned} v_\gamma^\pi(s) =& c(s, \pi(s)) + \gamma \mathbb{E}[v_\gamma^\pi(s')], \\ =& c(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s, a) v_\gamma^\pi(s'). \end{aligned} \tag{6}$$

The discount factor $\gamma \in [0, 1)$ limits the contribution of future rewards and $c(s, a)$ is defined in Equation (5).

5.2 Optimal Policy for Single-Unit System

The optimality equation (2) is assumed to return a stationary optimal policies for a stationary cost and transition function. An optimal policy $\pi^* : \mathcal{S} \to \mathcal{A}$, minimizes the expected discounted cumulative costs over the time horizon [46]:

$$\pi^* = \arg\min_\pi \mathbb{E}[\sum_{t=0}^{\infty} \gamma c(s_t, \pi(s_t), s_{t+1})] \tag{7}$$

This results in the following system of equations to determine the optimal policy π^*:

$$\pi^*(s) = \arg\min_\pi \{\gamma c(s, \pi(s)) + \sum_{s \in \mathcal{S}} v(s')\}, \quad s \in \mathcal{S}. \tag{8}$$

The optimal policy is expected to have a monotone structure, which is beneficial in their appeal to operators, the ease of implementation, and efficient computation. The existence of optimal monotone policies is shown under the following assumptions that ensure the existence of an optimal stationary lim inf policy[46]:

Assumption 5.1. *For each $s \in \mathcal{S}$, $-\infty < C \leq c(s, a) < \infty$.*

Assumption 5.2. *For each $s \in \mathcal{S}$, $0 \leq \gamma < 1$, $v_\gamma^*(s) < \infty$.*

Assumption 5.3. *There exists a $K < \infty$ such that, for each $s \in \mathcal{S}$, $h_\gamma^*(s) \equiv v_\gamma^*(s) - v_\gamma^*(0) \geq -K$ for $0 \leq \gamma < 1$.*

Assumption 5.4. *There exist a non-negative function $M(s)$ such that*

- ***a.*** *$M(s) < \infty$;*
- ***b.*** *for each $s \in \mathcal{S}$, $h_\gamma(s) \leq M(s)$ for all γ, $0 \leq \gamma < 1$; and*
- ***c.*** *for each $s \in \mathcal{S}$, and $a \in A_s$, $\sum_{s' \in S} p(s'|s, a) M(s') < \infty$.*

Further, monotone policies require the state to have a physical interpretation and a natural ordering. An ordering is a transitive, reflexive and antisymmetric relationship between elements in the set [46]:

Definition 5.1. Let W be an arbitrary set with a *partial ordering* on that set described by $\preceq$. For u, v, and $w \in W$, $u \preceq v, v \preceq w$ implies $u \preceq w$ (transitivity) and $w \preceq w$ (reflexivity), and $u \preceq v$ $v \preceq u$ implies $u = v$ (antisymmetry). A partially ordered set is called a *poset* [46].

Definition 5.2. Two elements u and v in W are *comparable* if either $u \preceq v$ or $v \preceq u$ [46].

Definition 5.3. A partial ordering is a *total ordering* on W if every pair of elements in W is comparable. A set with a total ordering is called a *chain* [46].

The natural ordering for each individual deterioration path in Sections 5.2.1-5.2.4 follows the deterioration from new to failure state. So $A_i \preceq A_j \preceq F_1, F_2$ for $i <\leq j$ and $F_1||F_2$. The orderings are similar for B, C, D, and E.

The following theorem from Puterman [46] gives conditions under which an optimal monotone policy exists. The first two conditions ensure the value function is nondecreasing in s and the last two ensure the policy is monotone in s. In the case of a two-action space, the monotone policy is a threshold policy. The stopping action becomes optimal when the units condition is sufficiently deteriorated. It is assumed that the action space is not a function of the state, so $\mathcal{A}_s = \mathcal{A}$ for all s [46].

Theorem 5.1. *Let $\mathcal{S} = \{0, 1, \ldots\}$, suppose that Assumptions 5.1-5.4 hold, and further that*

1. *$c(s, a)$ is nondecreasing in s for all $a \in \mathcal{A}$*
2. *$q(k|s, a) \equiv \sum_{s'=k}^{\infty} p(s'|s, a)$ is nondecreasing in s for all $k \in \mathcal{S}$ and $a \in \mathcal{A}$*
3. *$c(s, a)$ is a subadditive function on $\mathcal{S} \times \mathcal{A}$*
4. *$q(k|s, a)$ is a superadditive function on $\mathcal{S} \times \mathcal{A}$ for all $k \in \mathcal{S}$ alternative: $\sum_{s'=0}^{\infty} p(s'|s, a)u(s')$ is a superadditive function on $\mathcal{S} \times \mathcal{A}$ for nonincreasing u.*

Then there exists a lim inf average optimal stationary policy $(d^*)^\infty$ in which d^* is nondecreasing in s. Further, when $\mathcal{S}$ is finite, $(d^*)^\infty$ is average optimal.

The following lemma can be used to prove the sub- or superadditivity of functions.

Lemma 5.2. *Let $g(s, a)$ be a real-valued function on $\mathcal{S} \times \mathcal{A}$, with $\mathcal{A} = \{0, 1\}$ and $\mathcal{S} = \{0, 1, \ldots\}$. If $g(s, a)$ satisfies*

$$[g(s + 1, 1) - g(s + 1, 0)] - [g(s, 1) - g(s, 0)] \geq 0 \tag{9}$$

for all s, it is superadditive. If the reverse inequality holds, then $g(s, a)$ is said to be subadditive [46].

Further, Kallenberg [26] showed that under conditions 1 and 2 from Theorem 5.1, $v_\gamma(s)$ is nondecreasing in s. This is used to proof Assumption 5.3.

Lemma 5.3. *If*

- *$c(s, a)$ is nondecreasing in s for all $a \in \mathcal{A}$,*
- *$q(k|s, a) \equiv \sum_{s'=k}^{\infty} p(s'|s, a)$ is nondecreasing in s for all $k \in \mathcal{S}$ and $a \in \mathcal{A}$.*

Then value function $v_\gamma(s)$ is nondecreasing in $s \in \mathcal{S}$.

We show the structure for each dimension in the state space (as if each deterioration path evolves independently from the other paths).

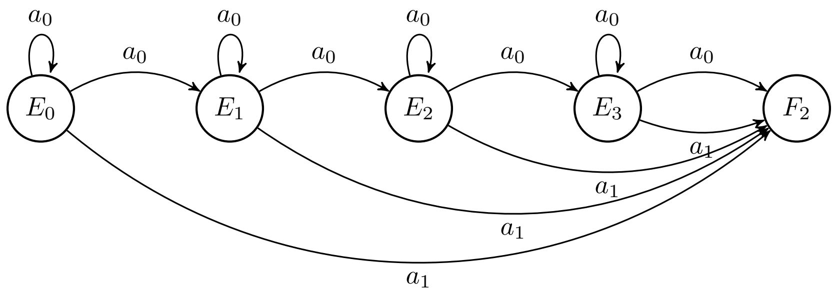
5.2.1 State Dimension E: Corrosion

Corrosion has a monotone increasing path that ends in a RUN DOWN with no risk for a CRASH. We expect that choosing the stopping action is not optimal in any state.

Define $\mathcal{S}_E = \{E_0, E_1, E_2, E_3, F_2\}$. At each decision epoch, the possible actions are to continue a_0 or to stop a_1, but the failure state requires no action. Let $s + 1 = E_{i+1}$ if $s = E_i$ for $i = 0, 1, 2$ and $s + 1 = F_2$ if $s = E_3$. The transition probabilities are

$$p(s'|s, a) = \begin{cases} 0, & s' \prec s, \\ p(s'|s), & s', \in \{s, s + 1, F_2\}, \quad s \in \{E_0, E_1, E_2, E_3\}, \quad a = a_0, \\ 1, & s' = F_2, \quad s \in \{E_0, E_1, E_2, E_3\}, \quad a = a_1. \end{cases}$$

This corresponds with the following absorbing Markov chain:



The edge weights denote the possible transitions corresponding to a certain action. Since $P(F_1|s) = 0$ for all s, the cost function is

$$c(s,a) = \begin{cases} -c_1(s), & a = a_0, \\ 0, & a = a_1. \end{cases} \tag{10}$$

Let $v^*_\gamma(s) = \min\limits_{a\in\mathcal{A}}\{c(s,a) + \gamma \sum_{s'} p(s'|s,a)v^*(s')\}$. Set $v(F_2) = Z$ for some constant $0 \leq Z < \infty$.

Theorem 5.4. *An optimal policy π^*_E exists, which is a threshold policy with no threshold:*

$$\pi^*_E(s) = \begin{cases} a_0, & s \in \{E_0, E_1, E_2, E_3\}, \\ a_1, & s \in \emptyset. \end{cases}$$

Proof. If Assumptions 5.1-5.4 and conditions 1-4 of Theorem 5.1 hold, then an optimal policy exists and is a threshold-type:

Assumption 5.1: Reward $c_1(s)$ is by definition nonincreasing with s and $0 < c_1(s) < \infty$. Therefore, $-c_1(E_0) \leq c(s,a) \leq 0$ and Assumption 5.1 holds with $C = -c_1(E_0)$.

Assumption 5.2: Let d^∞ denote the stationary policy to stop in every state: $d(s) = a_1$ for all s.

$$\begin{aligned} v^*_\gamma(s) \leq & v^d_\gamma(s) \\ =& 0 + \gamma \sum_{s'\in\mathcal{S}_E} p(s'|s,a_1) v_{\gamma\gamma}{}^d(s') \\ =& 0 + \gamma v^d_\gamma(F_2) \\ =& \gamma Z \\ <& \infty \end{aligned}$$

So Assumption 5.2 holds.

Assumption 5.3: Later we show that conditions 1 and 2 from Theorem 5.1 hold. Then by Lemma 5.3, $v^*_\gamma(s)$ does not decrease with s and $v^*_\gamma(s) \geq v^*_\gamma(E_0)$ for all s.

So that

$$\begin{aligned} h^*_\gamma(s) =& v^*_\gamma(s) - v^*_\gamma(E_0) \\ \geq& v^*_\gamma(E_0) - v^*_\gamma(E_0) \\ =& 0 \end{aligned}$$

So Assumption 5.3 holds with $K = 0$.

Assumption 5.4: Let $M(s) < \infty$, recall $v^*_\gamma(E_0) \leq v^*_\gamma(s) \leq 0 \leq v^*_\gamma(F_2) = Z$, for $s \in \mathcal{S}_E \setminus \{F_2\}$ and $\mathcal{A} = \{a_0, a_1\}$.

Then for $s \in \mathcal{S}_E$

$$\begin{aligned} h_\gamma(s) =& v^*_\gamma(s) - v^*_\gamma(E_0) \\ \leq& Z - v^*_\gamma(E_0) \\ =& Z - \min_{a\in A}\{c(0,a) + \gamma \sum_{s'\in\mathcal{S}_E} p(s'|E_0,a)v^*(s')\} \\ =& Z - \min\{\gamma Z, -c_1(E_0) + \gamma \sum_{s'} p(s'|E_0)v^*(s')\} \end{aligned}$$

So Assumption 5.4b holds for $M(s) = M = Z - \min\{\gamma Z, c_1(E_0) + \gamma \sum_{s'\in\mathcal{S}_E} p(s'|E_0)v^*(s')\} = Z - v^*_\gamma(E_0)$, which is well-defined (finite).

$$\begin{aligned} \sum_{s'\in\mathcal{S}_E} p(s'|s,a)M(s') =& \sum_{s'\in\mathcal{S}_E} p(s'|s,a)M \\ =& \sum_{s'\in\mathcal{S}_E} p(s'|s,a)(Z - v^*_\gamma(E_0)) \\ =& \Big(Z - v^*_\gamma(E_0)\Big) \sum_{s'\in S} p(s'|s,a) \\ =& Z - v^*_\gamma(E_0) \\ <& \infty \end{aligned}$$

Further, the conditions for Theorem 5.1 hold:

1. By definition $c(s,a)$ is nondecreasing in s for all $a \in \mathcal{A}$;
2. For $a = a_1$, $q(k|s,a)$ is independent of s, so nondecreasing. For $a = a_0$, define $\Delta q(k,s) \equiv q(k|s+1,a_0) - q(k|s,a_0)$. Then

$$\Delta q(k,s) = \begin{cases} \sum_{s'\in\mathcal{S}_E}(p(s'|s+1) - p(s'|s)), & k \succ s, \\ 0, & k \preceq s. \end{cases} \tag{11}$$

For $k = s+1$, the first term can be reduced to

$$\begin{aligned} \sum_{s'\succeq s+1, s'\in\mathcal{S}_E} (p(s'|s+1) - p(s'|s)) =& 1 - p(s+1|s) \\ \geq& 0. \end{aligned}$$

For $k = s+2$, the first term can be reduced to

$$\begin{aligned} \sum_{s'\succeq s+2, s'\in\mathcal{S}_E} (p(s'|s+1) - p(s'|s)) =& p(s+2|s+1) \\ \geq& 0. \end{aligned}$$

For $k \succeq s+3$, the first term can be reduced to

$$\begin{aligned} \sum_{s'\succeq s+3, s'\in\mathcal{S}_E} (p(s'|s+1) - p(s'|s)) =& 0 - 0 \\ =& 0. \end{aligned}$$

So $\Delta q(k,s)$ is nonnegative, implying $q(k|s,a)$ is nondecreasing in s for all $a \in \mathcal{A}$.

3.

$$\begin{aligned} [c(s+1,a_1) - c(s+1,a_0)] - [c(s,a_1) - c(s,a_0)] =& [0 - (-c_1(s+1))] - [0 - (-c_1(s))] \\ =& c_1(s+1) - c_1(s) \\ \leq& 0 \end{aligned}$$

The last inequality follows from $c_1(s)$ being nonincreasing in s. So $c(s,a)$ is subadditive by Lemma 5.2.

4.

$$\left[\sum_{s'\in\mathcal{S}_E} p(s'|s+1,a_1)u(s') - \sum_{s'\in\mathcal{S}_E} p(s'|s+1,a_0)u(s')\right] - \left[\sum_{s'\in\mathcal{S}_E} p(s'|s,a_1)u(s') - \sum_{s'\in\mathcal{S}_E} p(s'|s,a_0)u(s')\right]$$

$$\begin{aligned}
&= u(F_2) - \sum_{s'\in\mathcal{S}_E} p(s'|s+1,a_0)u(s') - \left[u(F_2) - \sum_{s'\in\mathcal{S}_E} p(s'|s,a_0)u(s')\right] \\
&= \sum_{s'\in\mathcal{S}_E} p(s'|s,a_0)u(s') - \sum_{s'\in\mathcal{S}_E} p(s'|s+1,a_0)u(s') \\
&= [p(s|s)u(s+1) + p(s+1|s)u(s)] - [p(s+1|s+1)u(s+1) + p(s+2|s+1)u(s+2)] \\
&\geq p(s|s)u(s+1) + p(s+1|s)u(s+1) - (p(s+1|s+1)u(s+1) + p(s+2|s+1)u(s+1)) \\
&= u(s+1) - u(s+1) \\
&= 0
\end{aligned}$$

The inequality follows from $u(s)$ being nonincreasing in s. So $\sum_{s'\in\mathcal{S}_E} p(s'|s,a)u(s')$ is a superadditive function by Lemma 5.2.

A RUN DOWN failure incurs no penalty, eliminating the trade-off between production rewards and the CRASH risk. From the cost function (Equation 10), we get $c(s,a_0) \leq 0$ and $c(s,a_1) = 0$. So the corresponding value function (that we want to minimize) is nonnegative for $s \in \{E_0, E_1, E_2, E_3\}$ and $a \in \{a_0, a_1\}$:

$$v_\gamma^\pi(s) = c(s,\pi(s)) + \sum_{s'\in\mathcal{S}_E} \gamma p(s'|s,a)v(s') = \begin{cases} -c_1(s) + \sum_{s'\in\mathcal{S}_E} \gamma v(s') \leq v(F_2), & \pi(s) = a_0, \\ v(F_2), & \pi(s) = a_1. \end{cases}$$

Hence, stopping (a_1) never returns a lower cost than continuing operation (a_0). The optimal policy π_E^* is to let the unit operate until it runs down by itself:

$$\pi_E^*(s) = \begin{cases} a_0, & s \in \{E_0, E_1, E_2, E_3\}, \\ a_1, & s \in \emptyset. \end{cases}$$

□

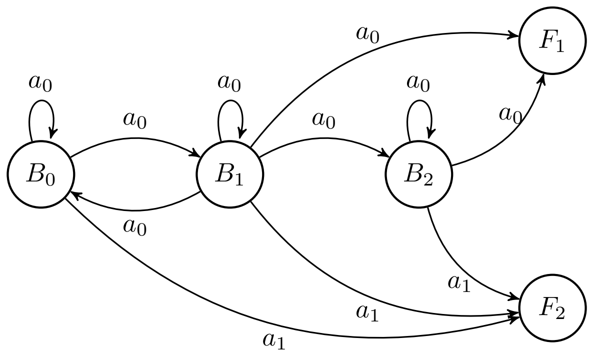
5.2.2 State Dimension B: Condensing Feed Impurities

Condensing Feed Impurities has a non-monotone path that ends in a CRASH. We expect that there will be a threshold for choosing the stopping action.

Define $\mathcal{S}_B = \{B_0, B_1, B_2, F_1\}$. At each decision epoch, the actions are to continue a_0 or to stop a_1, but the failure state requires no action. The transition probabilities are

$$p(s'|s,a) = \begin{cases} p(s'|s), & s' = s \in \{B_0, B_1, B_2\}, \quad a = a_0, \\ & \vee \quad s' = s+1, \quad s \in \{B_0, B_1, B_2, \}, \quad a = a_0, \\ & \vee \quad s' = B_0, \quad s = B_1, \quad a = a_0, \\ & \vee \quad s' = F_1, \quad s \in \{B_1, B_2\}, \quad a = a_0, \\ 1, & s' = F_2, \quad s \in \{B_0, B_1, B_2\}, \quad a = a_1, \\ 0, & \text{otherwise.} \end{cases}$$

This corresponds with the following Markov chain:



The cost is given by

$$c(s,a) = \begin{cases} -c_1(s) + C_3 p(F_1|s), & a = a_0, \\ 0, & a = a_1. \end{cases} \tag{12}$$

Further, the value function satisfies $v^*_\gamma(s) = \min_{a\in A}\{c(s,a) + \gamma \sum_{s'} p(s'|s,a)v^*_\gamma(s')\}$, and set $v_\gamma(F_1) = Z$ and $v_\gamma(F_2) = Z$ for some constant $0 \leq Z < \infty$.

Theorem 5.5. *An optimal policy π^*_B exists. This is a threshold policy with threshold τ_B.*

$$\pi^*_B(s) = \begin{cases} a_0, & s \prec \tau_B, \\ a_1, & s \succeq \tau_B. \end{cases}$$

Proof. Assumptions 5.1-5.4 hold:

Assumption 5.1: For $a = a_1$, $c(s,a) = 0$ for all s. For $a = a_0$, we have $c(s,a) = -c_1(s) + C_3P(F_1|s)$. Here, reward $c_1(s)$ is by definition nonincreasing with s, C_3 is a constant and $0 < c_1(s) \leq C_3 < \infty$. We assume the failure rate is increasing, so $P(F_1|s)$ increases with s. Function $c(s,a)$ is increasing with s, and $-c_1(B_0) < c(s,a) \leq C_3$. So Assumption 5.1 holds with $C = -c_1(B_0)$.

Assumption 5.2 We refer to the proof of Theorem 5.4.

Assumption 5.3 We refer to the proof of Theorem 5.4

Assumption 5.4 Let $M(s) < \infty$, recall $v^*_\gamma(B_0) \leq v^*_\gamma(s)$.

$$\begin{aligned} h_\gamma(s) =& v^*_\gamma(s) - v^*_\gamma(B_0) \\ =& \min_{a\in A}\{c(s,a) + \gamma \sum_{s'\in\mathcal{S}_B} p(s'|s,a)v^*(s')\} - \min_{a\in A}\{c(B_0,a) + \gamma \sum_{s'\in\mathcal{S}_B} p(s'|B_0,a)v^*(s')\} \\ =& \min\{\gamma Z, -c_1(s) + C_3 p(F_1|s) + \gamma \sum_{s'\in\mathcal{S}_B} p(s'|s)v^*(s')\} - \min\{\gamma Z, -c_1(B_0) + \gamma \sum_{s'\in\mathcal{S}_B} p(s'|B_0)v^*(s')\} \\ \leq& \gamma Z - \min\{\gamma Z, -c_1(B_0) + \gamma \sum_{s'\in\mathcal{S}_B} p(s'|B_0)v^*(s')\} \\ \leq& Z - \min\{\gamma Z, -c_1(B_0) + \gamma \sum_{s'\in\mathcal{S}_B} p(s'|B_0)v^*(s')\} \\ <& \infty \end{aligned}$$

So Assumption 5.4b holds for $M(s) = M = Z - \min\{0, -c_1(B_0) + \gamma \sum_{s'\in\mathcal{S}_B} p(s'|B_0)v^*(s')\} = Z - v^*_\gamma(B_0)$. Lastly, we refer to Theorem 5.4 for Assumption 5.4c.

Further, the conditions for Theorem 5.1 hold:

1. By definition $c(s,a)$ is nondecreasing in s for all $a \in \mathcal{A}$.

2. For $a = a_1$, $q(k|s,a)$ is independent of s, so nondecreasing. For $a = a_0$, define $\Delta q(k,s) \equiv q(k|s+1,a_0) - q(k|s,a_0)$. Then

$$\Delta q(k,s) = \sum_{s' \succeq k, s' \in \mathcal{S}_B} (p(s'|s+1) - p(s'|s)) \tag{13}$$

- Suppose $k = B_0$, then

$$\begin{aligned}\sum_{s' \succeq k, s' \in \mathcal{S}_B} (p(s'|s+1) - p(s'|s)) =& \sum_{s' \in \mathcal{S}_B} (p(s'|s+1) - p(s'|s)) \\ =&1 - 1 \\ =&0\end{aligned}$$

- Suppose $k = B_1$, then

$$\begin{aligned}\sum_{s' \succeq k, s' \in \mathcal{S}_B} (p(s'|s+1) - p(s'|s)) =& \sum_{s' \geq B_1, s' \in \mathcal{S}_B} (p(s'|s+1) - p(s'|s)) \\ =&[1 - p(B_0|s+1)] - [1 - p(B_0|s)] \\ =&p(B_0|s) - p(B_0|s+1)\end{aligned}$$

  We assume

$$p(B_0|B_0) \geq p(B_0|B_1),$$

  as a unit is more likely to stay in B_0 than to return to B_0 once it has experienced a significant amount of impurities B_1. So for $s = B_0$, $\Delta q(k,s)$ is nonnegative. For $s = B_1$, it reduces to

$$p(B_0|B_1) - p(B_0|B_2) = p(B_0|B_1) - 0 \geq 0.$$

  For $s \succeq B_2$, it reduces to zero.

- Suppose $k = B_2$, then

$$\begin{aligned}\sum_{s' \succeq k, s' \in \mathcal{S}_B} (p(s'|s+1) - p(s'|s)) =& \sum_{s' \succeq B_2, s' \in \mathcal{S}_B} (p(s'|s+1) - p(s'|s)) \\ =&[p(B_2|s+1) + p(F_1|s+1)] - [p(B_2|s) + p(F_1|s)]\end{aligned}$$

  For $s = B_0$, it reduces to

$$[p(B_2|B_1) + p(F_1|B_1)] - [p(B_2|B_0) + p(F_1|B_0)] = p(B_2|B_1) + [p(F_1|B_1) - p(F_1|B_0)] \geq 0.$$

  For $s = B_1$, it reduces to

$$[p(B_2|B_2) + p(F_1|B_2)] - [p(B_2|B_1) + p(F_1|B_1)] = 1 - p(B_2|B_1) \geq 0.$$

  For $s = B_2$, it reduces to

$$[p(B_2|F_1) + p(F_1|F_1)] - [p(B_2|B_2) + p(F_1|B_2)] = 1 - 1 = 0.$$

- Suppose $k = F_1$, then

$$\begin{aligned}\sum_{s' \succeq k, s' \in \mathcal{S}_B} (p(s'|s+1) - p(s'|s)) =&(p(F_1|s+1) - p(F_1|s)) \\ \geq&0\end{aligned}$$

So $\Delta q(k,s)$ is nonnegative in s, implying $q(k|s,a)$ is nondecreasing in s for all $a \in \mathcal{A}$.

3.

$$\begin{aligned}
&[c(s+1,a_1)-c(s+1,a_0)]-[c(s,a_1)-c(s,a_0)]\\
=&[0-(-c_1(s+1)+C_3p(F_1|s+1))]-[0-(-c_1(s)+C_3p(F_1|s)]\\
=&c_1(s+1)-C_3p(F_1|s+1)-c_1(s)+C_3p(F_1|s)\\
=&[c_1(s+1)-c_1(s)]+C_3[p(F_1|s)-p(F_1|s+1)]\\
\leq&0
\end{aligned}$$

The last inequality follows from $c_1(s)$ being nonincreasing in s and an increasing failure rate. So $c(s,a)$ is subadditive by Lemma 5.2.

4.

$$\begin{aligned}
&\left[\sum_{s'\in\mathcal{S}_B}p(s'|s+1,a_1)u(s')-\sum_{s'\in\mathcal{S}_B}p(s'|s+1,a_0)u(s')\right]-\left[\sum_{s'\in\mathcal{S}_B}p(s'|s,a_1)u(s')-\sum_{s'\in\mathcal{S}_B}p(s'|s,a_0)u(s')\right]\\
=&u(F_2)-\sum_{s'\in\mathcal{S}_B}p(s'|s+1,a_0)u(s')-\left[u(F_2)-\sum_{s'\in\mathcal{S}_B}p(s'|s,a_0)u(s')\right]\\
=&\sum_{s'\in\mathcal{S}_B}p(s'|s)u(s')-\sum_{s'\in\mathcal{S}_B}p(s'|s+1)u(s')
\end{aligned}$$

- Suppose $s=B_0$, then $s+1=B_1$ and

$$\begin{aligned}
&\sum_{s'\in\mathcal{S}_B}p(s'|s)u(s')-\sum_{s'\in\mathcal{S}_B}p(s'|s+1)u(s')\\
=&u(B_0)p(B_0|B_0)+u(B_1)p(B_1|B_0)\\
&\quad-[u(B_0)p(B_0|B_1)+u(B_1)p(B_1|B_1)+u(B_2)p(B_2|B_1)+u(F_1)p(F_1|B_1)]\\
\geq&u(B_0)[p(B_0|B_0)-p(B_0|B_1)]+u(B_1)p(B_1|B_0)-u(B_1)\left[p(B_1|B_1)+p(B_2|B_1)+p(F_1|B_1)\right]\\
=&u(B_0)[p(B_0|B_0)-p(B_0|B_1)]+u(B_1)p(B_1|B_0)-u(B_1)\left[1-p(B_0|B_1)\right]\\
=&u(B_0)[p(B_0|B_0)-p(B_0|B_1)]+u(B_1)\left[p(B_1|B_0)-1+p(B_0|B_1)\right]\\
=&u(B_0)[p(B_0|B_0)-p(B_0|B_1)]+u(B_1)\left[1-p(B_0|B_0)-1+p(B_0|B_1)\right]\\
=&u(B_0)[p(B_0|B_0)-p(B_0|B_1)]+u(B_1)\left[p(B_0|B_1)-p(B_0|B_0)\right]\\
\geq&u(B_1)[p(B_0|B_0)-p(B_0|B_1)]-u(B_1)[p(B_0|B_1)-p(B_0|B_0)]\\
=&0
\end{aligned}$$

  The inequalities follow from $u(s)$ being nonincreasing in s.

- Suppose $s=B_1$, then

$$\begin{aligned}
&\sum_{s'\in\mathcal{S}_B}p(s'|s)u(s')-\sum_{s'\in\mathcal{S}_B}p(s'|s+1)u(s')=\\
&u(B_0)p(B_0|B_1)+u(B_1)p(B_1|B_1)+u(B_2)p(B_2|B_1)u(F_1)p(F_1|B_1)\\
&\quad-[u(B_2)p(B_2|B_2)+u(F_1)p(F_1|B_2)]\\
=&u(B_0)p(B_0|B_1)+u(B_1)p(B_1|B_1)+u(B_2)[p(B_2|B_1)-p(B_2|B_2)]+u(F_1)[p(F_1|B_1)-p(F_1|B_2)]\\
\geq&u(B_2)p(B_0|B_1)+u(B_2)p(B_1|B_1)+u(B_2)[p(B_2|B_1)-p(B_2|B_2)]+u(F_1)[p(F_1|B_1)-p(F_1|B_2)]\\
=&u(B_2)\left[p(B_0|B_1)+p(B_1|B_1)+p(B_2|B_1)-p(B_2|B_2)\right]+u(F_1)[p(F_1|B_1)-p(F_1|B_2)]\\
=&u(B_2)\left[1-p(F_1|B_1)-[1-p(F_1|B_2)]\right]+u(F_1)[p(F_1|B_1)-p(F_1|B_2)]\\
=&u(B_2)\left[p(F_1|B_2)-p(F_1|B_1)\right]+u(F_1)[p(F_1|B_1)-p(F_1|B_2)]\\
\geq&u(F_1)\left[p(F_1|B_2)-p(F_1|B_1)\right]+u(F_1)[p(F_1|B_1)-p(F_1|B_2)]\\
=&0
\end{aligned}$$

- Suppose $s = B_2$, then

$$\begin{aligned}&\sum_{s'\in\mathcal{S}_B} p(s'|s)u(s') - \sum_{s'\in\mathcal{S}_B} p(s'|s+1)u(s') =\\ =&u(B_2)p(B_2|B_2) + u(F_1)p(F_1|B_2) - u(F_1)p(F_1|F_1)\\ \geq&u(F_1)p(B_2|B_2) + u(F_1)p(F_1|B_2) - u(F_1)\\ =&u(F_1) - u(F_1)\\ =&0\end{aligned}$$

So $\sum_{s'\in\mathcal{S}_B} p(s'|s,a)u(s')$ is a superadditive function by Lemma 5.2.

So if $p(B_0|B_0) \geq p(B_0|B_1)$, then the optimal policy π_B^* is a threshold policy with threshold τ_B.

$$\pi_B^*(s) = \begin{cases} a_0, & s \prec \tau_B, \\ a_1, & s \succeq \tau_B. \end{cases}$$

□

The policy states that at the threshold τ_B, the crash risk penalty outweighs production rewards, making it cost-optimal to stop the unit before a crash occurs.

5.2.3 State Dimension D: Low Temperature

We refer to Section 5.2.2 since both paths have the same number of states and transitions (replace 'B' with 'D').

Theorem 5.6. *An optimal policy π_D^* exists. This is a threshold policy with threshold τ_D:*

$$\pi_D^*(s) = \begin{cases} a_0 & s \prec \tau_D, \\ a_1 & s \succeq \tau_D. \end{cases}$$

Proof. For the proof we refer to the proof of Theorem 5.5 in Section 5.2.2. We assume that $p(D_0 \geq D_0) \geq p(D_0|D_1)$ as a unit is more likely stay in D_0 (normal temperature), than to transition back to D_0 once a unit experiences a lower temperature D_1. □

5.2.4 State Dimension C: Light Gas

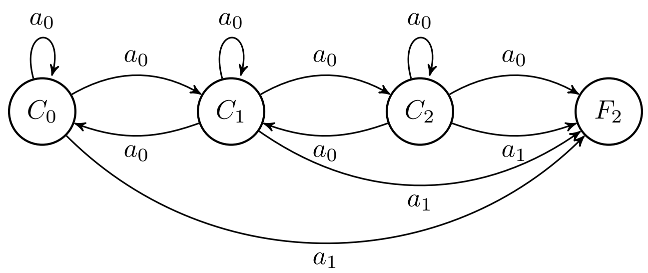
Light Gas has a non-monotone path that ends in a RUN DOWN. We expect that choosing the stopping action is not optimal in any state.

At each decision epoch, the actions are to continue a_0 or to stop a_1. Define $\mathcal{S}_C = \{C_0, C_1, C_2, C_3, F_2\}$.

The transition probabilities are

$$p(s'|s,a) = \begin{cases} p(s'|s), & s' = s \in \{C_0, C_1, C_2\}, \quad a = a_0 \\ & \vee s' = s+1, \quad s \in \{C_0, C_1, C_2, \}, \quad a = a_0 \\ & \vee s' = s-1, \quad s \in \{C_1, C_2\} \quad a = a_0, \\ 1, & s' = F_2, \quad s \in \{C_0, C_1, C_2\}, \quad a = a_1, \\ 0, & \text{otherwise.} \end{cases}$$

This corresponds with the following Markov chain:



Since $P(F_1|s) = 0$ for all s, the cost function is

$$c(s,a) = \begin{cases} -c_1(s), & a = a_0, \\ 0, & a = a_1. \end{cases} \tag{14}$$

Let $v^*_\gamma(s) = \min_{a\in\mathcal{A}}\{c(s,a)+\gamma\sum_{s'} p(s'|s,a)v^*(s')\}$, and set $v(F_2) = Z$ for some constant $0 \leq Z < \infty$.

Theorem 5.7. *An optimal policy π^*_C exists, which is a threshold policy with no threshold:*

$$\pi^*_C(s) = \begin{cases} a_0, & s \in \{C_0, C_1, C_2, C_3\}, \\ a_1 & s \in \emptyset. \end{cases}$$

Proof. We refer to Theorem 5.4 to show assumptions 5.1-5.4 hold, as well as conditions 1 and 3 from Theorem 5.1.

Conditions 2 and 4 from Theorem 5.1 also hold:

2. For $a = a_1$, $q(k|s,a)$ is independent of s, so nondecreasing. For $a = a_0$, define $\Delta q(k,s) \equiv q(k|s+1,a_0) - q(k|s,a_0)$. Then

$$\Delta q(k,s) = \begin{cases} \sum_{s'\succeq k, s'\in\mathcal{S}_C}(p(s'|s+1) - p(s'|s)) & k \succ s, \\ 0 & k \preceq s. \end{cases} \tag{15}$$

   We assume $p(C_0|C_0) \geq p(C_0|C_1)$ and $p(C_1|C_1) \geq p(C_1|C_2)$, as it is more likely to remain in a state than it is to transition back to a better state.

   - For $k = s+1$ and $s = C_0$, the first term can be reduced to

$$\begin{aligned}\sum_{s'\succeq C_1, s'\in\mathcal{S}_C}(p(s'|C_1) - p(s'|C_0)) =& p(C_1|C_1) - p(C_1|C_0) + p(C_2|C_1) \\ =& p(C_1|C_1) - (1 - p(C_0|C_0)) + p(C_2|C_1) \\ =& 1 - p(C_0|C_1) - (1 - p(C_0|C_0)) \\ =& p(C_0|C_0) - p(C_0|C_1) \\ \geq& 0\end{aligned}$$

   - For $k = s+1$ and $s = C_1$, the first term can be reduced to

$$\begin{aligned}\sum_{s'\succeq C_2, s'\in\mathcal{S}_C}(p(s'|C_2) - p(s'|C_1)) =& p(C_2|C_2) - p(C_2|C_1) + p(F_1|C_2) \\ =& 1 - p(C_1|C_2) - p(C_2|C_1) \\ \geq& 1 - p(C_1|C_1) - p(C_2|C_1) \\ =& p(C_0|C_1) \\ \geq& 0\end{aligned}$$

   - For $k = s+1$ and $s = C_2$, the first term can be reduced to

$$\begin{aligned}\sum_{s'\succeq C_3, s'\in\mathcal{S}_C}(p(s'|F_1) - p(s'|C_2)) =& p(F_1|F_1) - p(F_1|C_2) \\ =& 1 - p(F_1|C_2) \\ \geq& 0\end{aligned}$$

   - For $k = s+2$, the first term can be reduced to

$$\begin{aligned}\sum_{s'\succeq s+2, s'\in\mathcal{S}_C}(p(s'|s+1) - p(s'|s)) =& p(s+2|s+1) \\ \geq& 0.\end{aligned}$$

We have $p(s'|s) = 0$ for all $s' \geq s+2$. So for $k \geq s+3$, $\Delta q(k|s,a) = 0$. So condition 3 holds.

4.
$$\left[\sum_{s'\in\mathcal{S}_C} p(s'|s+1,a_1)u(s') - \sum_{s'\in\mathcal{S}_C} p(s'|s+1,a_0)u(s')\right] - \left[\sum_{s'\in\mathcal{S}_C} p(s'|s,a_1)u(s') - \sum_{s'\in\mathcal{S}_C} p(s'|s,a_0)u(s')\right]$$
$$= \left[u(F_2) - \sum_{s'\in\mathcal{S}_C} p(s'|s+1)u(s')\right] - \left[u(F_2) - \sum_{s'\in\mathcal{S}_C} p(s'|s)u(s')\right]$$
$$= \sum_{s'\in\mathcal{S}_C} p(s'|s)u(s') - \sum_{s'\in\mathcal{S}_C} p(s'|s+1)u(s')$$

Suppose $s = C_0$, then $s+1 = C_1$. Recall $p(C_0|C_0) \geq p(C_0|C_1)$ by assumption, then

$$\begin{aligned}
&\sum_{s'\in\mathcal{S}_C} p(s'|s)u(s') - \sum_{s'\in\mathcal{S}_C} p(s'|s+1)u(s') \\
=&p(C_0|C_0)u(C_0) + p(C_1|C_0)u(C_1) - [p(C_0|C_1)u(C_0) + p(C_1|C_1)u(C_1) + p(C_2|C_1)u(C_2)] \\
=& \left[p(C_0|C_0) - p(C_0|C_1)\right]u(C_0) + \left[p(C_1|C_0) - p(C_1|C_1)\right]u(C_1) - p(C_2|C_1)u(C_2) \\
\geq& \left[p(C_0|C_0) - p(C_0|C_1)\right]u(C_1) + \left[p(C_1|C_0) - p(C_1|C_1)\right]u(C_1) - p(C_2|C_1)u(C_2) \\
=& \left[p(C_0|C_0) + p(C_1|C_0) - p(C_0|C_1) - p(C_1|C_1)\right]u(C_1) - p(C_2|C_1)u(C_2) \\
=& \left[1 - (1 - p(C_2|C_1))\right]u(C_1) - p(C_2|C_1)u(C_2) \\
=&p(C_2|C_1)u(C_1) - p(C_2|C_1)u(C_2) \\
\geq&0
\end{aligned}$$

The last inequality follows, because $u(s)$ is nonincreasing in s.
Suppose $s = C_1$, then $s+1 = C_2$ and

$$\begin{aligned}
&\sum_{s'\in\mathcal{S}_C} p(s'|s)u(s') - \sum_{s'\in\mathcal{S}_C} p(s'|s+1)u(s') \\
=&p(C_0|C_1)u(C_0) + p(C_1|C_1)u(C_1) + p(C_2|C_1)u(C_2) - [p(C_1|C_2)u(C_1) + p(C_2|C_2)u(C_2) + p(F_2|C_2)u(F_2)] \\
=&p(C_0|C_1)u(C_0) + \left[p(C_1|C_1) - p(C_1|C_2)\right]u(C_1) + \left[p(C_2|C_1) - p(C_2|C_2)\right]u(C_2) - p(F_2|C_2)u(F_2) \\
\geq&p(C_0|C_1)u(C_2) + \left[p(C_1|C_1) - p(C_1|C_2)\right]u(C_2) + \left[p(C_2|C_1) - p(C_2|C_2)\right]u(C_2) - p(F_2|C_2)u(F_2) \\
=& \left[p(C_0|C_1) + p(C_1|C_1) - p(C_1|C_2) + p(C_2|C_1) - p(C_2|C_2)\right]u(C_2) - p(F_2|C_2)u(F_2) \\
=& \left[1 - (1 - p(F_2|C_2)\right]u(C_2) - p(F_2|C_2)u(F_2) \\
=&p(F_2|C_2)u(C_2) - p(F_2|C_2)u(F_2) \\
\geq&0
\end{aligned}$$

The last inequality follows, because $u(s)$ is nonincreasing in s. Suppose $s = C_2$, then $s+1 = F_2$ and

$$\begin{aligned}
\sum_{s'\in\mathcal{S}_C} p(s'|s)u(s') - \sum_{s'\in\mathcal{S}_C} p(s'|s+1)u(s') =&p(C_1|C_2)u(C_1) + p(C_2|C_2)u(C_2) + p(F_2|C_2)u(F_2) - u(F_2) \\
\geq&p(C_1|C_2)u(F_2) + p(C_2|C_2)u(F_2) + p(F_2|C_2)u(F_2) - u(F_2) \\
=&u(F_2) - u(F_2) \\
=&0
\end{aligned}$$

A RUN DOWN failure incurs no penalty, eliminating the trade-off between production rewards and the CRASH risk. So the optimal policy π_C^* is to let the unit operate until it runs down by itself:

$$\pi_C^*(s) = \begin{cases} a_0, & s \in \{C_0, C_1, C_2, C_3\}, \\ a_1 & s \in \emptyset. \end{cases}$$

□

5.2.5 State Dimension A: High Pressure

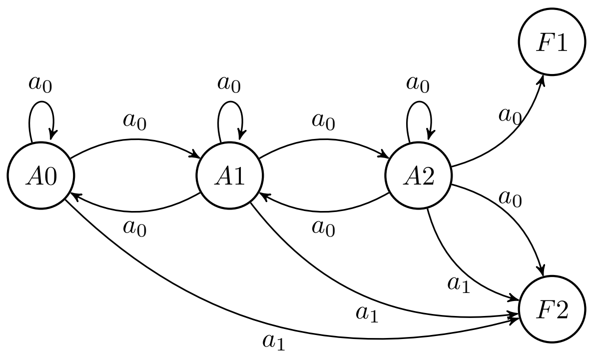
Condensing Feed Impurities has a non-monotone path that ends in a CRASH or RUN DOWN. We expect that there will be a threshold for choosing the stopping action.

At each decision epoch, the actions are to continue a_0 or to stop a_1. Define $\mathcal{S}_E = \{A_0, A_1, A_2, F_1, F_2\}$. The states A_0, A_1, A_2 are ordered with respect to the level of deterioration, but the path ends in either a CRASH F_1 or RUN DOWN F_2 (competing failure types). We propose that the ordering of states is not necessary for the absorbing states: $s = A_2$ gives $s+1 = F_1$ or $s+1 = F_2$.

The transition probabilities are

$$p(s'|s,a) = \begin{cases} p(s'|s), & s' = s \in \{A_0, A_1, A_2\}, \quad a = a_0 \\ & \vee s' = s+1, \quad s \in \{A_0, A_1, A_2\}, \quad s' \in \{A_1, A_2, F_1, F_2\} \quad a = a_0 \\ & \vee s' = s-1, \quad s \in \{A_1, A_2\} \quad a = a_0, \\ 1, & s' = F_2, \quad s \in \{A_0, A_1, A_2\}, \quad a = a_1, \\ 0, & \text{otherwise.} \end{cases}$$

This corresponds with the following Markov chain:



The cost function is

$$c(s,a) = \begin{cases} -c_1(s) + C_3 p(F_1|s), & a = a_0, \\ 0, & a = a_1. \end{cases} \tag{16}$$

The value function is $v_\gamma^*(s) = \min_{a \in \mathcal{A}} \{c(s,a) + \gamma \sum_{s'} p(s'|s,a) v^*(s')\}$. Set $v(F_1) = Z$ and $v(F_2) = Z$ for some constant $0 \leq Z < \infty$.

Theorem 5.8. *An optimal policy π_A^* exists. This is a threshold policy with threshold τ_A.*

$$\pi_A^*(s) = \begin{cases} a_0, & s \prec \tau_A, \\ a_1, & s \succeq \tau_A. \end{cases}$$

Proof. We refer to Theorem 5.5 to show that assumptions 5.1-5.4 hold, as well as conditions 1 and 3 from Theorem 5.1.

Conditions 2 and 4 from Theorem 5.1 also hold:

2. For $a = a_1$, $q(k|s,a)$ is independent of s, so nondecreasing. For $a = a_0$, define $\Delta q(k,s) \equiv q(k|s+1, a_0) - q(k|s, a_0)$. Then

$$\Delta q(k,s) = \begin{cases} \sum_{s' \succeq k, s' \in \mathcal{S}_A} (p(s'|s+1) - p(s'|s)) & k \succ s, \\ 0 & k \preceq s. \end{cases} \tag{17}$$

We refer to the proof of Theorem 5.7 for the cases that $k = s+1$ and $s = A_0$, $s = A_1$,

- For $k = s+1$ and $s = A_2$, when we take $s+1 = F_i$ with $i = 1, 2$ the first term can be reduced to

$$\sum_{s' \succeq F_i, s' \in \mathcal{S}_A} (p(s'|F_i) - p(s'|A_2)) = p(F_i|F_i) - p(F_i|A_2)$$
$$= 1 - P(F_i|A_1)$$
$$\geq 0$$

- For $k = s+2$, the first term reduces to zero for all s, since transitions do not skip a state.

4. We show that $\sum_{s' \in \mathcal{S}_A} p(s'|s,a)u(s')$ is a superadditive function by showing Lemma 5.2 holds:

$$\left[\sum_{s' \in \mathcal{S}_A} p(s'|s+1, a_1)u(s') - \sum_{s' \in \mathcal{S}_A} p(s'|s+1, a_0)u(s')\right] - \left[\sum_{s' \in \mathcal{S}_A} p(s'|s, a_1)u(s') - \sum_{s' \in \mathcal{S}_A} p(s'|s, a_0)u(s')\right]$$
$$= u(F_2) - \sum_{s' \in \mathcal{S}_A} p(s'|s+1, a_0)u(s') - \left[u(F_2) - \sum_{s' \in \mathcal{S}_A} p(s'|s, a_0)u(s')\right]$$
$$= \sum_{s' \in \mathcal{S}_A} p(s'|s)u(s') - \sum_{s' \in \mathcal{S}_A} p(s'|s+1)u(s')$$

- Suppose $s = A_0$, $s+1 = A_1$ and recall $p(A_0|A_0) \geq p(A_0|A_1)$ by assumption. Then

$$\sum_{s' \in \mathcal{S}_A} p(s'|s)u(s') - \sum_{s' \in \mathcal{S}_A} p(s'|s+1, a_0)u(s') =$$
$$= p(A_0|A_0)u(A_0) + p(A_1|A_0)u(A_1) - [p(A_0|A_1)u(A_0) + p(A_1|A_1)u(A_1) + p(A_2|A_1)u(A_2)]$$
$$= [p(A_0|A_0) - p(A_0|A_1)]\, u(A_0) + [p(A_1|A_0) - p(A_1|A_1)]\, u(A_1) - p(A_2|A_1)u(A_2)$$
$$\geq [p(A_0|A_0) - p(A_0|A_1)]\, u(A_1) + [p(A_1|A_0) - p(A_1|A_1)]\, u(A_1) - p(A_2|A_1)u(A_2)$$
$$= [p(A_0|A_0) - p(A_0|A_1) + p(A_1|A_0) - p(A_1|A_1)]\, u(A_1) - p(A_2|A_1)u(A_2)$$
$$= [1 - (1 - p(A_2|A_1))]\, u(A_1) + -p(A_2|A_1)u(A_2)$$
$$= p(A_2|A_1)u(A_1) - p(A_2|A_1)u(A_2)$$
$$\geq 0$$

- Suppose $s = A_1$, $s+1 = A_2$ and assume that $p(A_1|A_1) \geq p(A_1|A_2)$. Then

$$\sum_{s' \in \mathcal{S}_A} p(s'|s)u(s') - \sum_{s' \in \mathcal{S}_A} p(s'|s+1)u(s')$$
$$= p(A_0|A_1)u(A_0) + p(A_1|A_1)u(A_1) + p(A_2|A_1)u(A_2)$$
$$- [p(A_1|A_2)u(A_1) + p(A_2|A_2)u(A_2) + p(F_1|A_2)u(F_1) + p(F_2|A_2)u(F_2)]$$
$$= p(A_0|A_1)u(A_0) + [p(A_1|A_1) - p(A_1|A_2)]\, u(A_1) + [p(A_2|A_1) - p(A_2|A_2)]\, u(A_2)$$
$$- p(F_1|A_2)u(F_1) + p(F_2|A_2)u(F_2)$$
$$\geq p(A_0|A_1)u(A_2) + [p(A_1|A_1) - p(A_1|A_2)]\, u(A_2) + [p(A_2|A_1) - p(A_2|A_2)]\, u(A_2)$$
$$- p(F_1|A_2)u(F_1) - p(F_2|A_2)u(F_2)$$
$$= [p(A_0|A_1) + p(A_1|A_1) - p(A_1|A_2) + p(A_2|A_1) - p(A_2|A_2)]\, u(A_2) - p(F_1|A_2)u(F_1) - p(F_2|A_2)u(F_2)$$
$$= [1 - p(A_1|A_2) - p(A_2|A_2)]\, u(A_2) - p(F_1|A_2)u(F_1) - p(F_2|A_2)u(F_2)$$
$$= [1 - (1 - p(F_1|A_2) + p(F_2|A_2))]\, u(A_2) - p(F_1|A_2)u(F_1) - p(F_2|A_2)u(F_2)$$
$$= [p(F_1|A_2) + p(F_2|A_2)]\, u(A_2) - p(F_1|A_2)u(F_1) - p(F_2|A_2)u(F_2)$$
$$\geq [p(F_1|A_2) + p(F_2|A_2)]\, u(A_2) - [p(F_1|A_2) + p(F_2|A_2)] \max\{u(F_1), u(F_2)\}$$
$$\geq 0$$

The optimal policy π_A^* is a threshold policy with threshold τ_A:

$$\pi_A^*(s) = \begin{cases} a_0, & s \prec \tau_A, \\ a_1, & s \succeq \tau_A. \end{cases}$$

□

The policy states that at the threshold τ_A, the crash risk penalty outweighs production rewards, making it cost-optimal to stop the unit before a crash occurs.

5.2.6 Multi-Dimensional State

In sections 5.2.1-5.2.4, we considered each dimension of the state space $\mathcal{S}$ separately. But all deterioration paths evolve simultaneously, so now we determine the structure of the optimal policy when all dimensions are considered. The state $s \in \mathcal{S}$ with

$$\mathcal{S} = \mathcal{S}_A \times \mathcal{S}_B \times \mathcal{S}_C \times \mathcal{S}_D \times \mathcal{S}_E$$

is multi-dimensional. Here, $\mathcal{S}_i$, with $i = A, B, C, D, E$ are totally ordered sets as described in Sections 5.2.1-5.2.4. The Cartesian product $\mathcal{S} = \mathcal{S}_A \times \mathcal{S}_B \times \mathcal{S}_C \times \mathcal{S}_D \times \mathcal{S}_E$ is then a finite partially ordered set with a componentwise ordering.

Definition 5.4. Partial ordering operator $\preceq$ on an N-dimensional set S is defined as $s \preceq s'$, for any s, $s' \in S$ with $s = (s_1, s_2, \ldots, s_N)$ and $s = (s'_1, s'_2, \ldots, s'_N)$, if $s_i \preceq s'_i$ for all $i \in \{1, \ldots, N\}$. This is a *componentwise ordering* [42].

Definition 5.5. Given two partially ordered sets $(X, \preceq_X)$ and $(Y, \preceq_Y)$. A function $f : X \to Y$ is *monotone* (or *order-preserving*) if f preserves order; that is, for all $x, y, \in X$, if $x \preceq_X y$, then $f(x) \preceq_Y f(y)$. [12]

Theorem 5.9. *Let X be a partially ordered set with the property that any two chains in X of cardinal 2 have a common upper bound and a common lower bound; and let Y be any totally ordered set. If the function $f : X \to Y$ is monotonic on every chain (Definition 5.3) in X of cardinal 3, then f is monotonic on X [7].*

A special case of Theorem 5.9 is Proposition 5.1.

Proposition 5.1. *Let $X_1, \ldots, X_n$ be (non-empty) subsets of $\mathbb{R}$ and let $X = X_1 \times \ldots \times X_n$. If the function $f : X \to \mathbb{R}$ is monotonic on every chain in X of cardinal 3, then f is monotonic on X [7].*

We claim Theorem 5.10.

Theorem 5.10. *An optimal policy π^* exists. This is a monotone policy with a five-way threshold.*

Proof. We use Proposition 5.1 with $X_1 = A, X_2 = B, X_3 = C, X_4 = D, X_5 = E$, and $f = v_\gamma : \mathcal{S} \to \mathbb{R}$. In the proofs of Theorem 5.4- Theorem 5.8, it is shown that v_γ is monotonic on A, B, C, D, E, and hence v_γ is also monotonic on every chain in $X = X_1 \times X_2 \times X_3 \times X_4 \times X_5$. Additionally, Theorem 5.4- Theorem 5.8 showed that the optimal policies $\pi^* : \mathcal{S} \to \mathcal{A}$ are monotonic on $\mathcal{S}$. Action space $\mathcal{A} = \{a_0, a_1\}$ is totally ordered and can be mapped on $\mathbb{R}$. Then by Proposition 5.1, π^* is a monotone policy on $\mathcal{S}$.

The result is a 5-way threshold policy that is a 5-cube in the state space, where the stopping action a_1 is chosen if one of the edges is reached. □

This means that a critical state in a single deterioration path is automatically critical in the multi-dimensional state (regardless of the state of the other paths). For example, if $A_2 \succeq \tau_A$ so that the stopping action is chosen in A_2, then the stopping action is also chosen for $(A_2, B_j, C_k, D_l, E_m)$ regardless of j, k, l, m.

Our deterioration paths are dependent, so combination of deterioration states that may fall short in each of the single paths, could, in combination equal in weight to a full dimension. The stopping action would be chosen earlier. To illustrate how this work, we use an example.

Example

Consider paths D and E. The state space

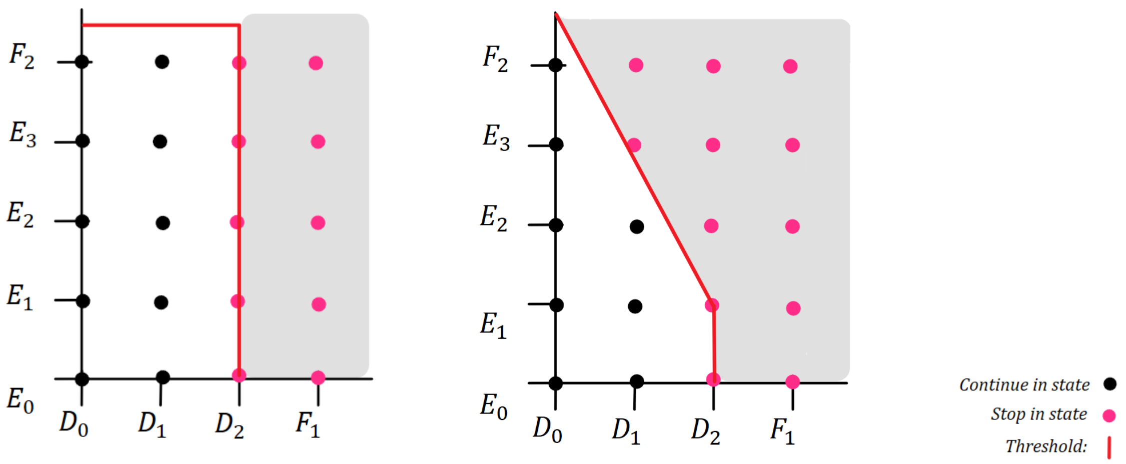
$$\mathcal{S}_D \times \mathcal{S}_E = \Big\{ \big\{ D_l, E_m) \quad \text{for} \quad l \in \{0,1,2\} \quad \text{and} \quad m \in \{0,1,2,3\} \big\} \cup \{F_1, F_2\} \Big\}$$

is two-dimensional. Theorems 5.6 and 5.4 state that the optimal policies for, respectively, path D and E are

$$\pi_D^*(s) = \begin{cases} a_0, & s \prec \tau_D, \quad s \in \mathcal{S}_D, \\ a_1, & s \succeq \tau_D, \quad s \in \mathcal{S}_D, \end{cases}$$

$$\pi_E^*(s) = \begin{cases} a_0, & s \in \mathcal{S}_E, \\ a_1, & s \in \emptyset. \end{cases}$$

Suppose $\tau_D = D_2$. If the paths evolve independently, then Figure 16a shows the two-way threshold, which is τ_D for dimension D and none for dimension E. Multiple states fall short of the threshold in their respective paths. However, if the paths evolve dependently (which is the case for our problem), then the states D_1 and E_3 could together be a critical condition where it would be optimal to stop. So the two-way threshold becomes a bound that is a combination of both thresholds, see Figure 16b.



(a) Two-way threshold for independent paths.

(b) Two-way threshold for dependent paths.

Figure 16

5.3 Policy Iteration for Monotone Policies

Policy iteration iterates over all policies and returns the optimal one. The monotone structure of the policy is used to effectively search through all policies and neglect certain suboptimal policies.

Puterman [46] describes a monotone policy iteration algorithm for totally ordered state space. We adapt this algorithm to make it applicable to partially ordered state spaces, see the pseudocode in Algorithm 1. The algorithm is structured as follows:

1. *Initialization*: Randomly choose a nondecreasing policy π.
2. *Policy Evaluation*: Bellman's equations are solved via an iterative scheme to approximate the value of the current policy.
3. *Policy Improvement*: We improve the current policy by greedily selecting actions that minimizes the value function. To guarantee the monotonicity of the new policy, we proceed as follows:
   - If action a_0 is chosen for state s, then assign action a_0 to all states that precede s.
   - If action a_1 is chosen for state s, then assign action a_1 to all states that succeed s.

   The policy evaluation and improvement steps are repeated until the policy no longer changes. During the policy improvement step the value of the policy can only decrease, therefore the policy converges to the optimal policy.

Algorithm 1 Policy Iteration Algorithm for a Monotone Policy with Partial Ordering

```
Input: γ: discount factor; θ: a small positive number; Z: a large positive number.
Output: π*: a deterministic optimal policy

// Initialization
Randomly initialize a nondecreasing policy π in s

stable-policy = False
while stable-policy = False do
    // Policy Evaluation
    π_old = π
    Δ = θ + 0.1
    while Δ > θ do
        Set v(F_1) = Z and v(F_2) = Z
        for s ∈ S \ {F_1, F_2} do
            v_old = v(s)
            v(s) = c(s, π_old(s)) + γ Σ_{s'} p(s'|s, π_old(s)) v(s')
            Δ = max{Δ, |v_old − v(s)|}
        end for
    end while

    // Policy Improvement
    stable-policy = True
    visited-states = emptylist
    for s ∈ S \ {F_1, F_2} do
        if s not in visited-states then
            Add s to visited-states
            π(s) = arg min_a {c(s, a) + γ Σ_{s'} p(s'|s, a) v(s')}
            if π(s) != π_old(s) then
                stable-policy = False
            end if
            if π(s) = continue then
                for s̃ preceding s that are not visited yet do
                    π(s̃) = continue
                    add s̃ to visited-states
                    if π(s̃) != π_old(s̃) then
                        stable-policy = False
                    end if
                end for
            else    // π(s) = stop
                for s̃ succeeding s that are not visited yet do
                    π(s̃) = stop
                    add s̃ to visited-states
                    if π(s̃) != π_old(s̃) then
                        stable-policy = False
                    end if
                end for
            end if
        end if
    end for
end while
return π* = π
```

5.4 MDP: Multi-Unit System

The production of the system decreases as more centrifuges fail. Non-operating centrifuges are replaced during refurbishments, which are planned when a significant increase in production can be made.

Refurbishments require preparations and cannot be executed immediately. Centrifuges should be in stock and workers should be available to install the centrifuges. Several centrifuges are always in stock, so we can assume 100% of a cascade is in stock. Urenco takes six to twelve months of preparation between deciding and executing a refurbishment. The duration of a refurbishment depends on the amount of centrifuges that need to be replaced and other maintenance activities. The total duration often spans multiple weeks. centrifuges that are still operating are not replaced in a refurbishment. If a centrifuge shows signs of some type of deterioration, it is relocated to a position at which that type of deterioration does not progress. Therefore, we state

Assumption 5.5. *Assume that after a refurbishment all units operate as new. So all units return to state* $(A_0, B_0, C_0, D_0, E_0)$.

5.4.1 Decision Epochs

The decision epochs are at discrete moments in time. The same intervals as in the single-unit model are used. Due to Assumption 5.5, the operation cycle ends once a refurbishment is executed. The system operates for a finite time before a refurbishment is planned, but the exact time is unknown. Therefore, we consider an infinite horizon $T = \{1, 2, \dots, \}$.

5.4.2 State variables

The state variable $s \in \mathcal{S}^M$ contains the deterioration state of all units. Each unit has the same possible states as in the single-unit model, so we also use state space $\mathcal{S}$ from the single-unit model (Section 5.1.2). Suppose we have m flomels, each containing n units. The state space is

$$\mathcal{S}^M = \{\mathbf{M}_{m\times n}, \quad \text{with} \quad \mathbf{M}_{i,j} \in \mathcal{S}, \quad \text{for} \quad i = 1, \dots, m, \quad j = 1, \dots, n\}.$$

5.4.3 Decision Variables

Two types of actions can be taken for any operating unit:

- a_0: *continue operating.* No action is taken, and there is no effect on the current operating state.
- a_1: *stop.* A single unit is run down.

The stochastic process of the multi-unit system continues when a single unit has failed. So we define one type of action for non-operating units:

- a_3: *continue non-operating.* No action is taken, and the unit remains non-operating.

Another action is taken on system level:

- a_2: *refurbishment.* A refurbishment is executed, that replaces all non-operating units.

Every decision epoch an action $a \in \mathcal{A}$ is to be taken where

$$\begin{aligned}
\mathcal{A} = \{\mathbf{A}^1, A^2\} \quad \text{with} \quad & \mathbf{A}^1_{i,j} \in \{a_0, a_1\}, && \text{if} \quad \mathbf{M}_{i,j} \notin \{F_1, F_2\}, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \\
& \mathbf{A}^1_{i,j} = a_3, && \text{if} \quad \mathbf{M}_{i,j} \in \{F_1, F_2\}, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \\
& A^2 \in \{a_0, a_2\}.
\end{aligned}$$

5.4.4 Transition Function

The transition function $p^M(s'|s)$ is the probability that the next state is $s' \in \mathcal{S}^M$ given the current state is $s \in \mathcal{S}^M$. It uses the transition function $p(s'|s)$ from the single-unit model in Section 5.1.4. Further, the factor $(1 + D_i)$ with $D_i \geq 0$ increases the probability of centrifuges in flomel i transitioning to a worse state due to a neighbouring CRASH. Each flomel has process and location neighbouring flomels (Section 1.1), so define factor d_L for when a CRASH occurred in a location-neighbouring flomel, where $N_{L,i}$ is the set of location-neighbouring flomels of flomel i, and d_P for when a CRASH occurred in a process-neighbouring flomel, where $N_{P,i}$ is the set of process-neighbouring flomels of flomel i.

Then

$$D_i = d_L \min\Big\{1, \sum_{k \in N_{L,i}} \sum_{j=1}^{n} \mathbb{1}\{\mathbf{M}_{k,j} = F_1\}\Big\} + d_P \min\Big\{1, \sum_{k \in N_{P,i}} \sum_{j=1}^{n} \mathbb{1}\{\mathbf{M}_{k,j} = F_1\}\Big\}.$$

To ensure the sum of probabilities equals one, the probability of transitioning to the same or a better state decreases with a factor $0 \leq Q_i \leq 1$, which follows from

$$\begin{aligned}
1 =& \frac{(1 + D_i) \sum_{M'_{i,j} \succ M_{i,j}} p(\mathbf{M}'_{i,j}|\mathbf{M}_{i,j}) + Q_i \sum_{M'_{i,j} \preceq M_{i,j}} p(\mathbf{M}'_{i,j}|\mathbf{M}_{i,j})}{(1 + D_i) \sum_{M'_{i,j} \succ M_{i,j}} p(\mathbf{M}'_{i,j}|\mathbf{M}_{i,j}) + Q_i \sum_{M'_{i,j} \preceq M_{i,j}} p(\mathbf{M}'_{i,j}|\mathbf{M}_{i,j})} \\
=& \frac{(1 + D_i)}{(1 + D_i) \sum_{M'_{i,j} \succ M_{i,j}} p(\mathbf{M}'_{i,j}|\mathbf{M}_{i,j}) + Q_i \sum_{M'_{i,j} \preceq M_{i,j}} p(\mathbf{M}'_{i,j}|\mathbf{M}_{i,j})} \sum_{M'_{i,j} \succ M_{i,j}} p(\mathbf{M}'_{i,j}|\mathbf{M}_{i,j}) \\
&+ \frac{Q_i}{(1 + D_i) \sum_{M'_{i,j} \succ M_{i,j}} p(\mathbf{M}'_{i,j}|\mathbf{M}_{i,j}) + Q_i \sum_{M'_{i,j} \preceq M_{i,j}} p(\mathbf{M}'_{i,j}|\mathbf{M}_{i,j})} \sum_{M'_{i,j} \preceq M_{i,j}} p(\mathbf{M}'_{i,j}|\mathbf{M}_{i,j}) \\
=& \alpha \sum_{M'_{i,j} \succ M_{i,j}} p(\mathbf{M}'_{i,j}|\mathbf{M}_{i,j}) + \beta \sum_{M'_{i,j} \preceq M_{i,j}} p(\mathbf{M}'_{i,j}|\mathbf{M}_{i,j}).
\end{aligned}$$

Here, we substituted the fractions by α and β for further ease of notation. Also, $1 \leq \alpha$ and $0 \leq \beta \leq 1$, this follows from $D \geq 0$.

For $s = \mathbf{M}$, $s' = \mathbf{M}'$, and $a = (\mathbf{A}^1, A^2)$

$$p^M(s'|s, a) = p^M(\mathbf{M}'_{1,1}|\mathbf{M}_{1,1}, a) \times \ldots \times p^M(\mathbf{M}'_{m,n}|\mathbf{M}_{m,n}, a)$$

with

$$p^M(\mathbf{M}'_{i,j}|\mathbf{M}_{i,j}, a) = \begin{cases} \alpha \cdot p(\mathbf{M}'_{i,j}|\mathbf{M}_{i,j}) & \mathbf{M}'_{i,j} \succ \mathbf{M}_{i,j}, \quad \mathbf{A}^1_{i,j} = a_0, \quad A^2 = a_0, \\ \beta \cdot p(\mathbf{M}'_{i,j}|\mathbf{M}_{i,j}) & \mathbf{M}'_{i,j} \preceq \mathbf{M}_{i,j}, \quad \mathbf{A}^1_{i,j} = a_0, \quad A^2 = a_0, \\ 1 & \mathbf{M}'_{i,j} = (A_0, B_0, C_0, D_0, E_0), \quad A^2 = a_2 \\ & \vee \quad \mathbf{M}'_{i,j} = F_2, \quad \mathbf{M}'_{i,j} \succ \mathbf{M}_{i,j}, \quad \mathbf{A}^1_{i,j} = a_1, \quad A^2 = a_0 \\ & \vee \quad \mathbf{M}_{i,j} = \mathbf{M}'_{i,j} \in \{F_1, F_2\}, \quad \mathbf{A}^1_{i,j} = a_3, \quad A^2 = a_0 \\ 0, & \text{otherwise.} \end{cases}$$

5.4.5 Cost Function

The cost function considers the production reward $c_{\text{product}}(s, a)$, replacement costs $c_{\text{replace}}(s, a)$ and penalties for crashes $c_{\text{crash}}(s, a, s')$ and non-operating units $c_{\text{non-operating}}(s, a)$:

$$\begin{aligned}
c(s, a) &= c_{\text{product}}(s, a) + c_{\text{replace}}(s, a) + c_{\text{crash}}(s, a) + c_{\text{non-operating}}(s, a) \qquad (18) \\
&= c_{\text{product}}(s, a) + c_{\text{replace}}(s, a) + \sum_{s' \in \mathcal{S}^M} c_{\text{crash}}(s, a, s') p(s'|s, a) + c_{\text{non-operating}}(s, a).
\end{aligned}$$

The total product reward is the sum of the product of all individual units per flomel. Similar to Section 5.1.5, the product and therefore rewards $c_1(M_{i,j})$ nonincrease with the partial ordering of the unit's state. The total reward is the sum over all centrifuges of the reward per centrifuge per time unit $c_1(M_{i,j}) \geq 0$, which is based on the unit's state.

$$c_{\text{product}}(s,a) = \begin{cases} \sum_{i=1}^{n} \sum_{j=1}^{m} -c_1(\mathbf{M}_{i,j}) \mathbb{1}\{\mathbf{A}^1_{i,j} = a_0\}, & A^2 = a_0, \\ 0, & A^2 = a_1. \end{cases}$$

The replacement costs $c_{\text{replace}}(s,a)$ represents the cost of a refurbishment. The cost includes the cost of hiring a crew and purchasing new units. Both components depend on the total number of units to be replaced. Therefore, we define a cost C_6 that combines the crew cost and the purchase cost of a new unit. Then

$$c_{\text{replace}}(s,a) = \begin{cases} 0, & A^2 = a_0, \\ \sum_{i=1}^{m} \sum_{j=1}^{n} \Big[\mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1, F_2\}\} + \mathbb{1}\{\mathbf{A}^1_{i,j} = a_1\} \Big] C_6, & A^2 = a_2. \end{cases}$$

We charge a fixed penalty C_3 for unexpected crashes, so $c_{\text{crash}}(s,a)$ depends on the next state $s' = \mathbf{M}'_{i,j}$:

$$c_{\text{crash}}(s,a) = \begin{cases} \sum_{i=1}^{m} \sum_{j=1}^{n} C_3 p(\mathbf{M}'_{i,j} = F_1 | \mathbf{M}_{i,j} \neq F_1) \mathbb{1}\{\mathbf{A}^1_{i,j} = a_0\}, & A^2 = a_0, \\ 0, & A^2 = a_2. \end{cases}$$

We introduce a penalty for every non-operating unit to account for the loss in production. This penalty has the same value as a crash penalty C_3, so that the cost function is monotone in the state, which is shown later in Section 5.5.

$$c_{\text{non-operating}}(s,a) = \begin{cases} \sum_{i=1}^{m} \sum_{j=1}^{n} C_3 \mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1, F_2\}\}, & A^2 = a_0, \\ 0, & A^2 = a_2. \end{cases}$$

Here, the penalty for a single CRASH $C_3$is higher than a unit's product reward $c_1(\mathbf{M}_{i,j})$ and all costs are finite. The production reward are defined as negative costs. So $0 \leq c_1(\mathbf{M}_{i,j}) \leq C_3 < \infty$ and $0 \leq C_6 < \infty$.

The cost function reduces to

$$c(s,a) = \begin{cases} c_{\text{product}}(s,a) + c_{\text{non-operating}}(s,a) + c_{\text{crash}}(s,a), & A^2 = a_0, \\ c_{\text{replace}}(s,a), & A^2 = a_2, \end{cases} \tag{19}$$

$$= \begin{cases} \sum_{i=1}^{n} \sum_{j=1}^{m} -c_1(\mathbf{M}_{i,j}) \mathbb{1}\{\mathbf{A}^1_{i,j} = a_0\} + C_3 \mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1, F_2\}\} \\ \quad + C_3 p(\mathbf{M}'_{i,j} = F_1 | \mathbf{M}_{i,j} \neq F_1) \mathbb{1}\{\mathbf{A}^1_{i,j} = a_0\}, & A^2 = a_0, \\ \sum_{i=1}^{m} \sum_{j=1}^{n} (\mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1, F_2\}\} + \mathbb{1}\{\mathbf{A}^1_{i,j} = a_1\}) C_6, & A^2 = a_2. \end{cases} \tag{20}$$

5.4.6 Value Function

The value function for policy π is the γ-discounted expected costs. For $s \in \mathcal{S}$,

$$\begin{aligned} v^{\pi}_{\gamma}(s) =& c(s, \pi(s)) + \gamma \mathbb{E}[v^{\pi}_{\gamma}(s')] \\ =& c(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} p^M(s'|s,a) v^{\pi}_{\gamma}(s') \end{aligned} \tag{21}$$

The discount factor $\gamma \in [0,1)$ limits the contribution of future rewards and $c(s,a)$ is defined in Equation (20).

5.5 Optimal Policy for the Multi-Unit System

To determine the structure of the optimal policy, we define the partial ordering of the state space as a componentwise ordering based on the condition of the units: Let $s, s' \in \mathcal{S}^M$, with $s = \mathbf{M}$ and $s' = \mathbf{M}'$. Then state s precedes state s', denoted as $s \preceq s'$, if $\mathbf{M}_{i,j} \preceq \mathbf{M}'_{i,j}$ for all $i = 1, \ldots, m$ and $j = 1, \ldots, n$.

Similarly, the partial ordering of the action space is also a componentwise ordering: Let $a, a^\star \in \mathcal{A}$, where $a = (\mathbf{A}^1, A^2)$ and $a^\star = (\mathbf{A}^{\star 1}, A^{\star 2})$. Then action a precedes action $a^\star$, denoted as $a \preceq a^\star$, if $\mathbf{A}^1_{i,j} \preceq \mathbf{A}^{\star 1}_{i,j}$ for all $i = 1, \dots, m$, $j = 1, \dots, n$ and $A^2 \preceq A^{\star 2}$, with the ordering $a_0 \preceq a_1 \preceq a_3$ and $a_0 \preceq a_2$.

Garcia et al. [18] introduce class-ordered monotone policies (CMP), that generalize monotone policies to cases where monotonicity holds on *ordered classes* of states and actions. Suppose the state space $\mathcal{S}^M$ is partitioned into ordered state classes $\mathcal{S}^M_1$, $\mathcal{S}^M_2$,..., $\mathcal{S}^M_G$ indexed by the set $\mathcal{G} = \{1, \dots, G\}$ and the action space $\mathcal{A}$ is partitioned into action classes $\mathcal{A}_1$, $\mathcal{A}_2$,..., $\mathcal{A}_H$ indexed by the set $\mathcal{H} = \{1, \dots, H\}$. Each state $s \in \mathcal{S}^M$ is mapped into one state class via function $\Theta : \mathcal{S}^M \to \mathcal{G}$ and each action is mapped into one action class via the function $\Phi : \mathcal{A} \to \mathcal{H}$. States and actions within a class are not ordered, and for any $g' > g$ and any state $s' \in \mathcal{S}^M_{g'}$ and state $s \in \mathcal{S}^M_g$, can be interpreted that state s' is more 'severe' than state s.

For the multi-unit model the action classes are composed as follows. Given action space $\mathcal{A}$, then actions $a = (\mathbf{A}^1, A^2)$ and $\tilde{a} = (\tilde{\mathbf{A}}^1, \tilde{A}^2)$ belong to the same action class, if they have the same refurbishment action A^2, and if $\mathbf{A}^1$, and $\tilde{\mathbf{A}}^1$ assign the same number of units to actions to a_0, a_1, and a_3. So if

$$A^2 = \tilde{A}^2,$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} \mathbb{1}\{\mathbf{A}^1_{i,j} = a_0\} = \sum_{i=1}^{m}\sum_{j=1}^{n} \mathbb{1}\{\tilde{\mathbf{A}}^1_{i,j} = a_0\},$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} \mathbb{1}\{\mathbf{A}^1_{i,j} = a_1\} = \sum_{i=1}^{m}\sum_{j=1}^{m} \mathbb{1}\{\tilde{\mathbf{A}}^1_{i,j} = a_1\},$$

and

$$\sum_{i=1}^{m}\sum_{j=1}^{n} \mathbb{1}\{\mathbf{A}^1_{i,j} = a_3\} = \sum_{i=1}^{m}\sum_{j=1}^{m} \mathbb{1}\{\tilde{\mathbf{A}}^1_{i,j} = a_3\}.$$

The partial ordering between the action classes is as follows: Suppose $a = (\mathbf{A}^1, A^2) \in \Phi$, and $\tilde{a} = (\tilde{\mathbf{A}}^1, \tilde{A}^2) \in \tilde{\Phi}$, then action class Φ precedes action class $\tilde{\Phi}$, denoted as $\Phi \preceq \tilde{\Phi}$, if

$$A^2 \preceq \tilde{A}^2,$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} \mathbb{1}\{\mathbf{A}^1_{i,j} = a_1\} \leq \sum_{i=1}^{m}\sum_{j=1}^{m} \mathbb{1}\{\tilde{\mathbf{A}}^1_{i,j} = a_1\},$$

and

$$\sum_{i=1}^{m}\sum_{j=1}^{n} \mathbb{1}\{\mathbf{A}^1_{i,j} = a_3\} \leq \sum_{i=1}^{m}\sum_{j=1}^{m} \mathbb{1}\{\tilde{\mathbf{A}}^1_{i,j} = a_3\}.$$

This is visualized for a system with two units in Figure 17. The action space is ordered in classes, where all actions within one class are incomparable. The ordering between classes is denoted by the operator $\preceq$.

The state classes are composed as follows: States $s = \mathbf{M}$ and $\tilde{s} = \tilde{\mathbf{M}}$ belong to the same class Θ if, for all $x \in \mathcal{S}$, states s and $\tilde{s}$ have the same number of units with state x. All units within a flomel are stochastically the same, but due to the factor D_i and $\tilde{D}_i$ in the transition function (Section 5.4.4) not all flomels are the same. Hence, these factors are also considered for the state classes:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} \mathbb{1}\{\mathbf{M}_{i,j} = x\}\mathbb{1}\{D_i = y\} = \sum_{i=1}^{m}\sum_{j=1}^{n} \mathbb{1}\{\tilde{\mathbf{M}}_{i,j} = x\}\mathbb{1}\{\tilde{D}_i = y\} \quad x \in \mathcal{S}, \quad y \in \{D_1, \dots, D_m, \tilde{D}_1, \dots, \tilde{D}_m\}.$$

The policy for a class-ordered state and actions space is defined as:

Definition 5.6. A policy π is a *class-ordered monotone policy* (CMP) if $\Theta(s) \geq \Theta(s')$ implies $\Phi(\pi(s)) \geq \Phi(\pi(s'))$ [18]

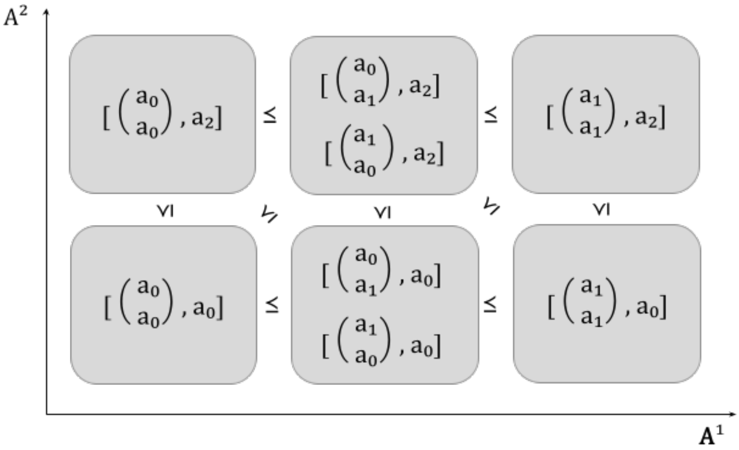


Figure 17: Classes for the action space $\mathcal{A} = (\mathcal{A}^1, A^2)$ with $\mathcal{A}^1_{2\times 1}$. The classes are represented by the rounded rectangles and ordered in dimensions $\mathcal{A}^1$ and A^2 according to the operator $\preceq$. Actions within a class are incomparable.

If Θ and Φ are the identity functions, so $\Theta(s) = s$ and $\Phi(a) = a$, then the resulting set of CMP's is the set of monotone policies. CMPs do not enforce strict monotonicity across states and actions, but they do retain the natural interpretability inherent in monotone policies.

Theorem 5.11. *The optimal policy is a class-ordered monotone policy decreasing in the class ordering of $\mathcal{S}$ and $\mathcal{A}$.*

Proof. Value function 21 is monotonically non-decreasing in s, if Assumptions 5.1- 5.4 hold and the four conditions of Theorem 5.1 in each dimension of the state and action space.

Assumption 5.1: *" For each $s \in \mathcal{S}^M$, $\quad -\infty < C \leq c(s,a) < \infty$."*
The cost function is

$$c(s,a) = c_{\text{product}}(s,a) + c_{\text{replace}}(s,a) + c_{\text{crash}}(s,a) + c_{\text{non-operating}}(s,a)$$

- Since $c_1(s)$ is by definition nonincreasing with s and finite, we have $0 \leq c_1(\mathbf{M}_{i,j}) \leq c_1((A_0, B_0, C_0, D_0, E_0))$, and the first component satisfies

$$-m \cdot n \cdot c_1((A_0, B_0, C_0, D_0, E_0)) \leq c_{\text{product}}(s,a) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_1(\mathbf{M}_{i,j})\mathbb{1}\{\mathbf{A}^1_{i,j} = a_0\} \leq 0 < \infty$$

- Cost component $c_{\text{replace}}(s,a)$ satisfies

$$0 \leq c_{\text{replace}}(s,a) \leq m \cdot n \cdot C_6 < \infty,$$

since at most $m \cdot n$ units can be replaced, and $0 \leq C_6 \leq \infty$ by definition.

- Clearly, $0 \leq p(\mathbf{M}'_{i,j} = F_1 | \mathbf{M}_{i,j} = F_1) \leq 1$, so that

$$0 \leq c_{\text{crash}}(s,a) \leq \sum_{i=1}^{m}\sum_{j=1}^{n} C_3 p(\mathbf{M}'_{i,j} = F_1 | \mathbf{M}_{i,j} \neq F_1)\mathbb{1}\{\mathbf{A}^1_{i,j} = a_0\} \leq m \cdot n \cdot C_3 < \infty.$$

- $0 \leq c_{\text{non-operating}}(s,a) \leq m \cdot n \cdot C_3 < \infty$

Therefore, $-m\cdot n\cdot c_1((A_0,B_0,C_0,D_0,E_0)) \le c(s,a) \le m\cdot n(C_6+2C_3) < \infty$ and Assumption 5.1 holds with $C = -n\cdot m\cdot c_1((A_0,B_0,C_0,D_0,E_0))$.

Assumption 5.2: *"For each $s \in \mathcal{S}^M$, $0 \le \gamma < 1$, $v^*_\gamma(s) < \infty$."*
Let d^∞ denote the stationary policy to stop every unit in every state and never plan a refurbishment: $\mathbf{A}_{i,j} = a_1$ for $i = 1,\dots,m$, $j = 1,\dots,n$, and $A^2 = a_0$. So $a = (\mathbf{a_1}, a_0)$. Then

$$\begin{aligned} v^*_\gamma(s) \le& v^d_\gamma(s) \\ =& c(s,a) + \gamma \sum_{s'} p^M(s'|s,a) v^d(s') \\ =& c_{\text{product}}(s,a) + c_{\text{crash}}(s,a) + c_{\text{non-operating}}(s,a) \\ =& 0 + 0 + m\cdot n\cdot C_3 \\ <& \infty \end{aligned}$$

So Assumption 5.2 holds.

Assumption 5.3: *"There exists a $K < \infty$ such that, for each $s \in \mathcal{S}^M$, $h^*_\gamma(s) \equiv v^*_\gamma(s) - v^*_\gamma(0) \ge -K$ for $0 \le \gamma < 1$."*
Conditions 1 and 2 from Theorem 5.1 (that are proven later) imply that the value function is monotone increasing in s. So

$$\begin{aligned} h^*_\gamma(s) =& v^*_\gamma(s) - v^*_\gamma(\mathbf{0}) \\ \ge& v^*_\gamma(\mathbf{0}) - v^*_\gamma(\mathbf{0}) \\ =& 0 \end{aligned}$$

Here, state $s = \mathbf{0}$ means $\mathbf{M}_{i,j} = (A_0,B_0,C_0,D_0,E_0)$ for $i = 1,\dots,m$ and $j = 1,\dots,n$. Assumption 5.3 holds for $K = 0$

Assumption 5.4: *"There exist a non-negative function $M(s)$ such that*

a. *$M(s) < \infty$;*
b. *for each $s \in \mathcal{S}$, $h_\gamma(s) \le M(s)$ for all γ, $0 \le \gamma < 1$; and*
c. *for each $s \in \mathcal{S}$, and $a \in A_s$, $\sum_{s' \in \mathcal{S}^M} p(s'|s,a)M(s') < \infty$."*

Let $M(s) < \infty$, recall $v^*_\gamma(s) \ge v^*_\gamma(\mathbf{0})$.

We assume no units are stopped a_1, while being in state (A_0,B_0,C_0,D_0,E_0), so that

$$\begin{aligned} c(\mathbf{0},a) =& c_{\text{product}}(\mathbf{0},a) + c_{\text{replace}}(\mathbf{0},a) + c_{\text{crash}}(\mathbf{0},a) + c_{\text{non-operating}}(\mathbf{0},a) \\ =& -\sum_{i=1}^{m} C_{\text{product}}(\mathbf{M}_{i,:}, \mathbf{A}^1_{i,:}) + 0 + 0 + 0 \\ =& -m\cdot n\cdot c_1((A_0,B_0,C_0,D_0,E_0)). \end{aligned}$$

Let d^∞ denote the stationary policy to stop every unit in every state and never plan a refurbishment: $d^\infty = (\mathbf{A}, A^2)$, with $\mathbf{A}_{i,j} = a_1$ for $i = 1,\dots,m$, $j = 1,\dots,n$, and $A^2 = a_0$. Then $v^d_\gamma(s) = m\cdot n\cdot C_3$.

$$\begin{aligned} h_\gamma(s) =& v^*_\gamma(s) - v^*_\gamma(\mathbf{0}) \\ \le& v^d_\gamma(s) - v^*_\gamma(\mathbf{0}) \\ \le& m\cdot n\cdot C_3 - \Big(-m\cdot n\cdot c_1((A_0,B_0,C_0,D_0,E_0)) + \gamma \sum_{s' \in \mathcal{S}^M} p^M(s'|\mathbf{0}) v^*(s')\Big) \\ =& m\cdot n\cdot C_3 + m\cdot n\cdot c_1((A_0,B_0,C_0,D_0,E_0)) - \gamma \sum_{s' \in \mathcal{S}^M} p^M(s'|\mathbf{0}) v^*(s') \\ \le& m\cdot n\cdot C_3 + m\cdot n\cdot c_1((A_0,B_0,C_0,D_0,E_0)) - \sum_{s' \in \mathcal{S}^M} v^*(s') \\ <& \infty \end{aligned}$$

The last inequality follows since $v*(s) < \infty$ by Assumption 5.3 and since the state space $\mathcal{S}^M$ is finite.

So let $M(s) = m \cdot n \cdot C_3 + m \cdot n \cdot c_1((A_0, B_0, C_0, D_0, E_0)) - \sum_{s'} v^*(s')$

$$\begin{aligned}\sum_{s' \in S} p(s'|s,a)M(s') = &\sum_{s' \in S} p(s'|s,a)M \\ =&M \sum_{s' \in S} p(s'|s,a) \\ =&M \\ <&\infty\end{aligned}$$

So Assumption 5.4 holds with $M = m \cdot n \cdot C_3 + m \cdot n \cdot c_1((A_0, B_0, C_0, D_0, E_0)) - \sum_{s'} v^*(s')$

Further, the conditions for Theorem 5.1 hold:

Conditions Theorem 5.1: → **1**, 2, 3, 4.

1. *"$c(s,a)$ is nondecreasing in s for all $a \in \mathcal{A}$"*

- Fix $A^2 = a_0$, then for any $\mathbf{A}^1$, the cost function is

$$\begin{aligned}c(s,a) = -\sum_{i=1}^{m}\sum_{j=1}^{n} &c_1(\mathbf{M}_{i,j})\mathbb{1}\{\mathbf{A}^1_{i,j} = a_0\} + C_3 p(\mathbf{M}'_{i,j} = F_1|\mathbf{M}_{i,j} \neq F_1)\mathbb{1}\{\mathbf{A}^1_{i,j} = a_0\} \\ &+ C_3\mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1, F_2\}\}\end{aligned}$$

  $-c_1(\mathbf{M}_{i,j})$ increases in s by definition. Due to an increasing failure rate $p(\mathbf{M}'_{i,j} = F_1|\mathbf{M}_{i,j} \neq F_1)$ increases in s till failure is reached. Upon reaching failure, c_{crash} becomes zero, but the loss in costs is compensated by $c_{\text{non-operating}}$. So $c(s,a)$ increases in s for $A^2 = a_0$.

- Fix $A^2 = a_2$, then for any $\mathbf{A}^1$, the cost function is

$$c(s,a) = \sum_{i=1}^{m}\sum_{j=1}^{n}\Big(\mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1, F_2\}\} + \mathbb{1}\{\mathbf{A}^1_{i,j} = a_1\}\Big)C_6$$

  This increases as more units reach a failure state, so $c(s,a)$ increases in s for $A^2 = a_2$.

Therefore, the cost function $c(s,a)$ increases in s for all actions $a \in \mathcal{A}$.

Conditions Theorem 5.1: 1, → **2**, 3, 4.

2. *"$q(k|s,a) \equiv \sum_{k \preceq s', s' \in \mathcal{S}^M} p(s'|s,a)$ is nondecreasing in s for all $k \in \mathcal{S}^M$ and $a \in \mathcal{A}$ "*
Define
$$\Delta q(k|s,a) = q(k|s+1,a) - q(k|s,a).$$

- For $A^2 = a_2$ and any $\mathbf{A}^1$, the operation cycle ends, and the system returns with probability one to an as-good-as-new new state: $s^* = \mathbf{M}^*$ with $M^*_{i,j} = (A_0, B_0, C_0, D_0, E_0)$ for $i = 1, \dots, m$, $j = 1, \dots, n$. Two cases can happen for k:

$$\begin{aligned}\Delta q(k|s,a) = &\sum_{k \preceq s'} p^M(s'|s+1,a) - \sum_{k \preceq s'} p^M(s'|s,a) \\ = &\begin{cases} p^M(s^*|s+1,a) - p^M(s^*|s,a) = 1 - 1 = 0, & \text{for} \quad k \preceq s^* \\ 0 - 0 = 0, & \text{for} \quad s^* \prec k \end{cases}\end{aligned}$$

  .

- For $A^2 = a_0$, and any $\mathbf{A}^1$, let $s = \mathbf{M}$, $s+1 = \mathbf{M}^\star$, $k = \mathbf{K}$. Then $s \preceq s+1$ implies for $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, n\}$, either

Conditions Theorem 5.1: 1, → **2**, 3, 4.

(a) $\mathbf{M}_{i,j} = \mathbf{M}^\star_{i,j}$;
(b) $\mathbf{M}_{i,j} \prec \mathbf{M}^\star_{i,j}$.
If $s = s+1$, then $\Delta q(k|s,a) = 0$. For $s \prec s+1$, at least one unit must satisfy case (b).

$$
\begin{aligned}
\Delta q(k|s,a) &= \sum_{k \prec s'} p^M(s'|s+1,a) - \sum_{k \prec s'} p^M(s'|s,a) \\
&= \sum_{k \prec s'} p^M(s'|s+1,a) - p^M(s'|s,a) \\
&= \sum_{\mathbf{K} \prec \mathbf{M}'} p^M(\mathbf{M}'|\mathbf{M}^\star,a) - p^M(\mathbf{M}'|\mathbf{M},a) \\
&= \sum_{\mathbf{K} \prec \mathbf{M}'} p^M(\mathbf{M}'_{1,1}|\mathbf{M}^\star_{1,1},a) \times \ldots \times p^M(\mathbf{M}'_{m,n}|\mathbf{M}^\star_{m,n},a) \\
&\quad - p^M(\mathbf{M}'_{1,1}|\mathbf{M}_{1,1},a) \times \ldots \times p^M(\mathbf{M}'_{m,n}|\mathbf{M}_{m,n},a)
\end{aligned}
$$

We reduce this further, by fixing all dimensions in the state s but one, so assume all units satisfy case a) and one satisfies case b). W.l.o.g. let this be unit $M_{1,1}$ We show $\Delta q(k|s,a) \geq 0$ in dimension $M_{1,1}$.

$$
\begin{aligned}
&\sum_{\mathbf{K} \prec \mathbf{M}'} p^M(\mathbf{M}'|\mathbf{M}^\star,a) - p^M(\mathbf{M}'|\mathbf{M},a) \\
\overset{(1)}{=} &\sum_{\mathbf{K}_{1,1} \prec \mathbf{M}'_{1,1}} p^M(\mathbf{M}'_{1,1}|\mathbf{M}^\star_{1,1},a) \times \ldots \times p^M(\mathbf{M}'_{m,n}|\mathbf{M}^\star_{m,n},a) \\
&- p^M(\mathbf{M}'_{1,1}|\mathbf{M}_{1,1},a) \times \ldots \times p^M(\mathbf{M}'_{m,n}|\mathbf{M}_{m,n},a) \\
= &\sum_{\mathbf{K}_{1,1} \prec \mathbf{M}'_{1,1}} p^M(\mathbf{M}'_{1,1}|\mathbf{M}^\star_{1,1},a) \times \ldots \times p^M(\mathbf{M}'_{m,n}|\mathbf{M}_{m,n},a) \\
&- p^M(\mathbf{M}'_{1,1}|\mathbf{M}_{1,1},a) \times \ldots \times p^M(\mathbf{M}'_{m,n}|\mathbf{M}_{m,n},a) \\
\overset{(2)}{=} &\sum_{\mathbf{K}_{1,1} \prec \mathbf{M}'_{1,1}} p^M(\mathbf{M}'_{1,1}|\mathbf{M}_{1,1},a) \times \Omega \\
&- p^M(\mathbf{M}'_{1,1}|\mathbf{M}_{1,1},a) \times \Omega \\
= \Omega &\sum_{\mathbf{K}_{1,1} \prec \mathbf{M}'_{1,1}} p^M(\mathbf{M}'_{1,1}|\mathbf{M}^\star_{1,1},a) - p^M(\mathbf{M}'_{1,1}|\mathbf{M}_{1,1},a) \\
= \Omega \Big(&\sum_{\mathbf{K}_{1,1} \prec \mathbf{M}'_{1,1} \prec \mathbf{M}_{1,1} \prec \mathbf{M}^\star_{1,1}} \beta_1 p(\mathbf{M}'_{1,1}|\mathbf{M}^\star_{1,1},a) - \beta_1 p(\mathbf{M}'_{1,1}|\mathbf{M}_{1,1},a) \\
+ &\sum_{\mathbf{K}_{1,1} \preceq \mathbf{M}_{1,1} \prec \mathbf{M}^\star_{1,1} \preceq \mathbf{M}'_{1,1}} \alpha_1 p(\mathbf{M}'_{1,1}|\mathbf{M}^\star_{1,1},a) - \alpha_1 p(\mathbf{M}'_{1,1}|\mathbf{M}_{1,1},a) \Big) \\
= \Omega \Big(&\sum_{\mathbf{K}_{1,1} \prec \mathbf{M}'_{1,1} \prec \mathbf{M}_{1,1} \prec \mathbf{M}^\star_{1,1}} \beta_1 \Big[p(\mathbf{M}'_{1,1}|\mathbf{M}^\star_{1,1},a) - p(\mathbf{M}'_{1,1}|\mathbf{M}_{1,1},a) \Big] \\
+ &\sum_{\mathbf{K}_{1,1} \preceq \mathbf{M}_{1,1} \prec \mathbf{M}^\star_{1,1} \preceq \mathbf{M}'_{1,1}} \alpha_1 \Big[p(\mathbf{M}'_{1,1}|\mathbf{M}^\star_{1,1},a) - p(\mathbf{M}'_{1,1}|\mathbf{M}_{1,1},a) \Big] \Big) \\
\geq \Omega \Big(&\sum_{\mathbf{K}_{1,1} \prec \mathbf{M}'_{1,1} \prec \mathbf{M}_{1,1} \prec \mathbf{M}^\star_{1,1}} \beta_1 \Big[p(\mathbf{M}'_{1,1}|\mathbf{M}^\star_{1,1},a) - p(\mathbf{M}'_{1,1}|\mathbf{M}_{1,1},a) \Big] \\
+ &\sum_{\mathbf{K}_{1,1} \preceq \mathbf{M}_{1,1} \prec \mathbf{M}^\star_{1,1} \preceq \mathbf{M}'_{1,1}} \beta_1 \Big[p(\mathbf{M}'_{1,1}|\mathbf{M}^\star_{1,1},a) - p(\mathbf{M}'_{1,1}|\mathbf{M}_{1,1},a) \Big] \Big)
\end{aligned}
$$

$$=\Omega \sum_{\mathbf{K}_{1,1} \prec \mathbf{M}'_{1,1}} \beta_1 \Big[p(\mathbf{M}'_{1,1} | \mathbf{M}^\star_{1,1}, a) - p(\mathbf{M}'_{1,1} | \mathbf{M}_{1,1}, a) \Big]$$

$$\overset{(3)}{=} 0$$

(1) Uses that $\mathbf{M}_{i,j} = \mathbf{M}_{i,j^\star}$ for $i = 1$, $j = 2, \ldots, n$ and $i = 2, \ldots, m$, for $j = 1, \ldots, n$. (2) $p^M(\mathbf{M}'_{1,2} | \mathbf{M}_{1,2}, a) \times \ldots \times p^M(\mathbf{M}'_{m,n} | \mathbf{M}_{m,n}, a)$ does not change within the sum, so substitute this term by the constant Ω. (3) Follows from the proofs from Theorems 5.4- 5.8. So $\Delta q(k|s,a) \geq 0$ in $M_{i,j}$ for all $i = 1, \ldots, m$, $j = 1, \ldots, n$. Therefore, $\Delta q(k|s,a) \geq 0$ in s.

Conditions Theorem 5.1: 1, 2, → **3**, 4.

3. *"$c(s,a)$ is a subadditive function on $\mathcal{S}^M \times \mathcal{A}$"*

- *In dimension* $\mathbf{A}^1$.
  Let A^2 be fixed, and suppose $s = \mathbf{M}$, $s+1 = \mathbf{M}^\star$, $a = (\mathbf{A}^1, A^2)$, $a^\star = (\mathbf{A}^{\star 1}, A^2)$, so that $a \prec a^\star$. Then, for $A^2 = a_0$:

$$\begin{aligned}
&[c(s+1,a^\star) - c(s+1,a)] - [c(s,a^\star) - c(s,a)] \\
=& \Big[\sum_{i=1}^m \sum_{j=1}^n -c_1(\mathbf{M}^\star_{i,j}) \mathbb{1}\{\mathbf{A}^{\star 1}_{i,j} = a_0\} + C_3 p({\mathbf{M}^\star}'_{i,j} = F_1 | \mathbf{M}^\star_{i,j} \neq F_1) \mathbb{1}\{\mathbf{A}^{\star 1}_{i,j} = a_0\} \\
&\quad + C_3 \mathbb{1}\{\mathbf{M}^\star_{i,j} \in \{F_1, F_2\}\} \\
&- \Big(\sum_{i=1}^m \sum_{j=1}^n -c_1(\mathbf{M}_{i,j}) \mathbb{1}\{\mathbf{A}^{\star 1}_{i,j} = a_0\} + C_3 p(\mathbf{M}'_{i,j} = F_1 | \mathbf{M}_{i,j} \neq F_1) \mathbb{1}\{\mathbf{A}^{\star 1}_{i,j} = a_0\} \\
&\quad + C_3 \mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1, F_2\}\} \Big) \Big] \\
&- \Big[\sum_{i=1}^m \sum_{j=1}^n -c_1(\mathbf{M}_{i,j}) \mathbb{1}\{\mathbf{A}^{\star 1}_{i,j} = a_0\} + C_3 p(\mathbf{M}'_{i,j} = F_1 | \mathbf{M}_{i,j} \neq F_1) \mathbb{1}\{\mathbf{A}^{\star 1}_{i,j} = a_0\} \\
&\quad + C_3 \mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1, F_2\}\} \\
&- \Big(\sum_{i=1}^m \sum_{j=1}^n -c_1(\mathbf{M}_{i,j}) \mathbb{1}\{\mathbf{A}^{1}_{i,j} = a_0\} + C_3 p(\mathbf{M}'_{i,j} = F_1 | \mathbf{M}_{i,j} \neq F_1) \mathbb{1}\{\mathbf{A}^{1}_{i,j} = a_0\} \\
&\quad + C_3 \mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1, F_2\}\} \Big) \Big] \\
=& \sum_{i=1}^m \sum_{j=1}^n \Big(c_1(\mathbf{M}_{i,j}) - c_1(\mathbf{M}^\star_{i,j}) \Big) \mathbb{1}\{\mathbf{A}^{\star 1}_{i,j} = a_0\} + \Big(c_1(\mathbf{M}^\star_{i,j}) - c_1(\mathbf{M}_{i,j}) \Big) \mathbb{1}\{\mathbf{A}^{1}_{i,j} = a_0\} \qquad (22) \\
&+ C_3 \Big(p({\mathbf{M}^\star}'_{i,j} = F_1 | \mathbf{M}^\star_{i,j} \neq F_1) - p(\mathbf{M}'_{i,j} = F_1 | \mathbf{M}_{i,j} \neq F_1) \Big) \mathbb{1}\{\mathbf{A}^{\star 1}_{i,j} = a_0\} \qquad (23) \\
&- C_3 \Big(p({\mathbf{M}^\star}'_{i,j} = F_1 | \mathbf{M}^\star_{i,j} \neq F_1) - p(\mathbf{M}'_{i,j} = F_1 | \mathbf{M}_{i,j} \neq F_1) \Big) \mathbb{1}\{\mathbf{A}^{1}_{i,j} = a_0\} \qquad (24)
\end{aligned}$$

  Now, $a \prec a^\star$, this brings three cases for $i = 1, \ldots, m$, $j = 1, \ldots, n$, either
  (a) $\mathbf{A}^1_{i,j} = a_0$, $\mathbf{A}^{\star 1}_{i,j} = a_0$;
  (b) $\mathbf{A}^1_{i,j} = a_0$, $\mathbf{A}^{\star 1}_{i,j} = a_1$;
  (c) $\mathbf{A}^1_{i,j} = a_1$, $\mathbf{A}^{\star 1}_{i,j} = a_1$.

  For each case, respectively, lines (23) and (24) reduce componentwise as follows:
  (a) Then $\mathbb{1}\{\mathbf{A}^{\star 1}_{i,j} = a_0\} = 1 = \mathbb{1}\{\mathbf{A}^{1}_{i,j} = a_0\}$, so lines (22), (23) and (24) reduce to

Conditions Theorem 5.1: 1, 2, → **3**, 4.

zero:

$$\begin{aligned}
&\Big(c_1(\mathbf{M}_{i,j}) - c_1(\mathbf{M}^\star_{i,j})\Big)\mathbb{1}\{\mathbf{A}^{\star 1}_{i,j} = a_0\} + \Big(c_1(\mathbf{M}^\star_{i,j}) - c_1(\mathbf{M}_{i,j})\Big)\mathbb{1}\{\mathbf{A}^1_{i,j} = a_0\} \\
&+ C_3\Big(p({\mathbf{M}^\star}'_{i,j} = F_1|\mathbf{M}^\star_{i,j} \neq F_1) - p(\mathbf{M}'_{i,j} = F_1|\mathbf{M}_{i,j} \neq F_1)\Big)\mathbb{1}\{\mathbf{A}^{\star 1}_{i,j} = a_0\} \\
&- C_3\Big(p({\mathbf{M}^\star}'_{i,j} = F_1|\mathbf{M}^\star_{i,j} \neq F_1) - p(\mathbf{M}'_{i,j} = F_1|\mathbf{M}_{i,j} \neq F_1)\Big)\mathbb{1}\{\mathbf{A}^1_{i,j} = a_0\} \\
=&c_1(\mathbf{M}_{i,j}) - c_1(\mathbf{M}^\star_{i,j}) + c_1(\mathbf{M}^\star_{i,j}) - c_1(\mathbf{M}_{i,j}) \\
&+ C_3\Big(p({\mathbf{M}^\star}'_{i,j} = F_1|\mathbf{M}^\star_{i,j} \neq F_1) - p(\mathbf{M}'_{i,j} = F_1|\mathbf{M}_{i,j} \neq F_1) - \\
&- \Big(p({\mathbf{M}^\star}'_{i,j} = F_1|\mathbf{M}^\star_{i,j} \neq F_1) - p(\mathbf{M}'_{i,j} = F_1|\mathbf{M}_{i,j} \neq F_1)\Big)\Big) \\
=&0
\end{aligned}$$

(b) Then $\mathbb{1}\{\mathbf{A}^1_{i,j} = a_0\} = 1$ and $\mathbb{1}\{\mathbf{A}^{\star 1}_{i,j} = a_0\} = 0$, so lines (22), (23) and (24) reduce to:

$$\begin{aligned}
&\Big(c_1(\mathbf{M}_{i,j}) - c_1(\mathbf{M}^\star_{i,j})\Big)\mathbb{1}\{\mathbf{A}^{\star 1}_{i,j} = a_0\} + \Big(c_1(\mathbf{M}^\star_{i,j}) - c_1(\mathbf{M}_{i,j})\Big)\mathbb{1}\{\mathbf{A}^1_{i,j} = a_0\} \\
&+ C_3\Big(p({\mathbf{M}^\star}'_{i,j} = F_1|\mathbf{M}^\star_{i,j} \neq F_1) - p(\mathbf{M}'_{i,j} = F_1|\mathbf{M}_{i,j} \neq F_1)\Big)\mathbb{1}\{\mathbf{A}^{\star 1}_{i,j} = a_0\} \\
&- C_3\Big(p({\mathbf{M}^\star}'_{i,j} = F_1|\mathbf{M}^\star_{i,j} \neq F_1) - p(\mathbf{M}'_{i,j} = F_1|\mathbf{M}_{i,j} \neq F_1)\Big)\mathbb{1}\{\mathbf{A}^1_{i,j} = a_0\} \\
=&c_1(\mathbf{M}^\star_{i,j}) - c_1(\mathbf{M}_{i,j}) - C_3\Big(p({\mathbf{M}^\star}'_{i,j} = F_1|\mathbf{M}^\star_{i,j} \neq F_1) - p(\mathbf{M}'_{i,j} = F_1|\mathbf{M}_{i,j} \neq F_1)\Big) \\
\leq&0
\end{aligned}$$

Where the last equality is the same as the equations in the proofs 5.2.1 and 5.2.2 of the single-unit model, hence the inequality follows.

(c) Then $\mathbb{1}\{\mathbf{A}^{\star 1}_{i,j} = a_0\} = 0 = \mathbb{1}\{\mathbf{A}^1_{i,j} = a_0\}$, so lines (23) and (24) reduce to zero:

$$\begin{aligned}
&\Big(c_1(\mathbf{M}_{i,j}) - c_1(\mathbf{M}^\star_{i,j})\Big)\mathbb{1}\{\mathbf{A}^{\star 1}_{i,j} = a_0\} + \Big(c_1(\mathbf{M}^\star_{i,j}) - c_1(\mathbf{M}_{i,j})\Big)\mathbb{1}\{\mathbf{A}^1_{i,j} = a_0\} \\
&+ C_3\Big(p({\mathbf{M}^\star}'_{i,j} = F_1|\mathbf{M}^\star_{i,j} \neq F_1) - p(\mathbf{M}'_{i,j} = F_1|\mathbf{M}_{i,j} \neq F_1)\Big)\mathbb{1}\{\mathbf{A}^{\star 1}_{i,j} = a_0\} \\
&- C_3\Big(p({\mathbf{M}^\star}'_{i,j} = F_1|\mathbf{M}^\star_{i,j} \neq F_1) - p(\mathbf{M}'_{i,j} = F_1|\mathbf{M}_{i,j} \neq F_1)\Big)\mathbb{1}\{\mathbf{A}^1_{i,j} = a_0\} \\
=&0 + 0 + 0 - 0 \\
=&0
\end{aligned}$$

For $A^2 = a_2$, and any $\mathbf{A}^1$:

$$\begin{aligned}
&[c(s+1, a^\star) - c(s+1, a)] - [c(s, a^\star) - c(s, a)] \\
=&\Big[\sum_{i=1}^{m}\sum_{j=1}^{n}(\mathbb{1}\{\mathbf{M}^\star_{i,j} \in \{F_1, F_2\}\} + \mathbb{1}\{\mathbf{A}^{\star 1}_{i,j} = a_1\})C_6 \\
&- \Big(\sum_{i=1}^{m}\sum_{j=1}^{n}(\mathbb{1}\{\mathbf{M}^\star_{i,j} \in \{F_1, F_2\}\} + \mathbb{1}\{\mathbf{A}^1_{i,j} = a_1\})C_6\Big)\Big] \\
&- \Big[\sum_{i=1}^{m}\sum_{j=1}^{n}(\mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1, F_2\}\} + \mathbb{1}\{\mathbf{A}^{\star 1}_{i,j} = a_1\})C_6
\end{aligned}$$

Conditions Theorem 5.1:
1,
2,
→ **3**,
4.

$$
\begin{aligned}
&- \Big(\sum_{i=1}^{m}\sum_{j=1}^{n}(\mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1,F_2\}\} + \mathbb{1}\{\mathbf{A}^1_{i,j}=a_1\})C_6\Big)\Big] \\
=&\sum_{i=1}^{m}\sum_{j=1}^{n} C_6\Big(\mathbb{1}\{\mathbf{M}^\star_{i,j} \in \{F_1,F_2\}\} + \mathbb{1}\{\mathbf{A}^{\star 1}_{i,j}=a_1\}) \\
&- \mathbb{1}\{\mathbf{M}^\star_{i,j} \in \{F_1,F_2\}\} - \mathbb{1}\{\mathbf{A}^1_{i,j}=a_1\} \\
&- \mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1,F_2\}\} - \mathbb{1}\{\mathbf{A}^{\star 1}_{i,j}=a_1\} \\
&+ \mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1,F_2\}\} + \mathbb{1}\{\mathbf{A}^1_{i,j}=a_1\}\Big) \\
=&0.
\end{aligned}
$$

- *In dimension A^2.*
Let $\mathbf{A}^1$ be fixed, and suppose $s = \mathbf{M}$, $s+1 = \mathbf{M}^\star$, $a = (\mathbf{A}^1, a_0)$, $a^\star = (\mathbf{A}^1, a_2)$, so that $a \prec a^\star$. Then

$$
\begin{aligned}
&[c(s+1,a^\star) - c(s+1,a)] - [c(s,a^\star) - c(s,a)] \\
=&\Big[\sum_{i=1}^{m}\sum_{j=1}^{n}(\mathbb{1}\{\mathbf{M}^\star_{i,j} \in \{F_1,F_2\}\} + \mathbb{1}\{\mathbf{A}^1_{i,j}=a_1\})C_6 - \Big(\sum_{i=1}^{m}\sum_{j=1}^{n} -c_1(\mathbf{M}^\star_{i,j})\mathbb{1}\{\mathbf{A}^1_{i,j}=a_0\} \\
&+ C_3 p({\mathbf{M}^\star}'_{i,j} = F_1 | \mathbf{M}^\star_{i,j} \neq F_1)\mathbb{1}\{\mathbf{A}^1_{i,j}=a_0\} + C_3\mathbb{1}\{\mathbf{M}^\star_{i,j} \in \{F_1,F_2\}\}\Big)\Big] \\
&- \Big[\sum_{i=1}^{m}\sum_{j=1}^{n}(\mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1,F_2\}\} + \mathbb{1}\{\mathbf{A}^1_{i,j}=a_1\})C_6 - \Big(\sum_{i=1}^{m}\sum_{j=1}^{n} -c_1(\mathbf{M}_{i,j})\mathbb{1}\{\mathbf{A}^1_{i,j}=a_0\} \\
&+ C_3 p(\mathbf{M}'_{i,j} = F_1 | \mathbf{M}_{i,j} \neq F_1)\mathbb{1}\{\mathbf{A}^1_{i,j}=a_0\} + C_3\mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1,F_2\}\}\Big)\Big] \\
=&\sum_{i=1}^{m}\sum_{j=1}^{n}(\mathbb{1}\{\mathbf{M}^\star_{i,j} \in \{F_1,F_2\}\} - \mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1,F_2\}\})C_6 + (c_1(\mathbf{M}^\star_{i,j}) - c_1(\mathbf{M}_{i,j}))\mathbb{1}\{\mathbf{A}^1_{i,j}=a_0\} \\
&+ C_3 p(\mathbf{M}'_{i,j} = F_1 | \mathbf{M}_{i,j} \neq F_1)\mathbb{1}\{\mathbf{A}^1_{i,j}=a_0\} + C_3 p({\mathbf{M}^\star}'_{i,j} = F_1 | \mathbf{M}^\star_{i,j} \neq F_1)\mathbb{1}\{\mathbf{A}^1_{i,j}=a_0\} \\
&- C_3\mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1,F_2\}\} - C_3\mathbb{1}\{\mathbf{M}^\star_{i,j} \in \{F_1,F_2\}\} \\
=&\sum_{i=1}^{m}\sum_{j=1}^{n}(\mathbb{1}\{\mathbf{M}^\star_{i,j} \in \{F_1,F_2\}\} - \mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1,F_2\}\})C_6 + (c_1(\mathbf{M}^\star_{i,j}) - c_1(\mathbf{M}_{i,j}))\mathbb{1}\{\mathbf{A}^1_{i,j}=a_0\} \\
&+ C_3\Big(p(\mathbf{M}'_{i,j} = F_1 | \mathbf{M}_{i,j} \neq F_1)\mathbb{1}\{\mathbf{A}^1_{i,j}=a_0\} + \mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1,F_2\}\} \\
&- p({\mathbf{M}^\star}'_{i,j} = F_1 | \mathbf{M}^\star_{i,j} \neq F_1)\mathbb{1}\{\mathbf{A}^1_{i,j}=a_0\} - \mathbb{1}\{\mathbf{M}^\star_{i,j} \in \{F_1,F_2\}\}\Big)
\end{aligned}
$$

  - In case $\mathbf{A}^1_{i,j} = a_1$, assume $\mathbf{M}_{i,j}, \mathbf{M}^\star_{i,j} \notin \{F_1, F_2\}$ since no $A^1_{i,j} = a_0$ once a unit (i,j) has reached a failure state.

$$
\begin{aligned}
&(\mathbb{1}\{\mathbf{M}^\star_{i,j} \in \{F_1,F_2\}\} - \mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1,F_2\}\})C_6 + (c_1(\mathbf{M}^\star_{i,j}) - c_1(\mathbf{M}_{i,j}))\mathbb{1}\{\mathbf{A}^1_{i,j}=a_0\} \\
&+ C_3\Big(p(\mathbf{M}'_{i,j} = F_1 | \mathbf{M}_{i,j} \neq F_1)\mathbb{1}\{\mathbf{A}^1_{i,j}=a_0\} - \mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1,F_2\}\} \\
&- p({\mathbf{M}^\star}'_{i,j} = F_1 | \mathbf{M}^\star_{i,j} \neq F_1)\mathbb{1}\{\mathbf{A}^1_{i,j}=a_0\} - \mathbb{1}\{\mathbf{M}^\star_{i,j} \in \{F_1,F_2\}\}\Big) \\
=&0 + 0 + C_3\Big(p(\mathbf{M}'_{i,j} = F_1 | \mathbf{M}_{i,j} \neq F_1) - p({\mathbf{M}^\star}'_{i,j} = F_1 | \mathbf{M}^\star_{i,j} \neq F_1)\Big) \\
\geq&0
\end{aligned}
$$

  Due to an increasing failure rate.

  - In case $\mathbf{A}^1_{i,j} = a_0$, we distinguish three cases:

(a) $\mathbf{M}_{i,j}, \mathbf{M}^\star_{i,j} \notin \{F_1, F_2\}$, so that

$$\begin{aligned}&(\mathbb{1}\{\mathbf{M}^\star_{i,j} \in \{F_1,F_2\}\} - \mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1,F_2\}\})C_6 + (c_1(\mathbf{M}^\star_{i,j}) - c_1(\mathbf{M}_{i,j}))\mathbb{1}\{\mathbf{A}^1_{i,j} = a_0\}\\ &+ C_3\Big(p(\mathbf{M}'_{i,j} = F_1|\mathbf{M}_{i,j} \neq F_1)\mathbb{1}\{\mathbf{A}^1_{i,j} = a_0\} + \mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1,F_2\}\}\\ &- p(\mathbf{M}^{\star\prime}_{i,j} = F_1|\mathbf{M}^\star_{i,j} \neq F_1)\mathbb{1}\{\mathbf{A}^1_{i,j} = a_0\} - \mathbb{1}\{\mathbf{M}^\star_{i,j} \in \{F_1,F_2\}\}\Big)\\ =&c_1(\mathbf{M}^\star_{i,j}) - c_1(\mathbf{M}_{i,j})\\ &+ C_3\Big(p(\mathbf{M}'_{i,j} = F_1|\mathbf{M}_{i,j} \neq F_1) - p(\mathbf{M}^{\star\prime}_{i,j} = F_1|\mathbf{M}^\star_{i,j} \neq F_1)\Big)\\ \leq&0\end{aligned}$$

Since $c_1(\mathbf{M}_{i,j})$ decreases with the unit state $\mathbf{M}_{i,j}$ and the failure rate increases.

(b) $\mathbf{M}_{i,j} \notin \{F_1, F_2\}, \mathbf{M}^\star_{i,j} \in \{F_1, F_2\}$, then

$$\begin{aligned}&(1-0)C_6 + (0 - c_1(\mathbf{M}_{i,j}))\\ &+ C_3\Big(p(\mathbf{M}'_{i,j} = F_1|\mathbf{M}_{i,j} \neq F_1) + 0\\ &- 0 - 1\Big)\\ =&C_6 - c_1(\mathbf{M}_{i,j}) + C_3\Big(p(\mathbf{M}'_{i,j} = F_1|\mathbf{M}_{i,j} \neq F_1) - 1\end{aligned}$$

Now $-c_1(\mathbf{M}_{i,j}) \leq 0$, and $0 \leq p(\mathbf{M}'_{i,j} = F_1|\mathbf{M}_{i,j} \neq F_1) \leq 1$, so the second and third component are nonpositive. The first component, C_6 is positive and a constant. Since $A^2 = a_0$ and $A^{\star 2} = a_2$, assume $\sum_{i=1}^m \sum_{j=1}^n \Big(C_6 - c_1(\mathbf{M}_{i,j})\Big)\mathbb{1}\{\mathbf{M}_{i,j} \notin \{F_1,F_2\}, \mathbf{M}^\star_{i,j} \in \{F_1,F_2\}\} \leq 0$

(c) $\mathbf{M}_{i,j}, \mathbf{M}^\star_{i,j} \in \{F_1, F_2\}$, then

$$\begin{aligned}&(\mathbb{1}\{\mathbf{M}^\star_{i,j} \in \{F_1,F_2\}\} - \mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1,F_2\}\})C_6 + (c_1(\mathbf{M}^\star_{i,j}) - c_1(\mathbf{M}_{i,j}))\mathbb{1}\{\mathbf{A}^1_{i,j} = a_0\}\\ &+ C_3\Big(p(\mathbf{M}'_{i,j} = F_1|\mathbf{M}_{i,j} \neq F_1)\mathbb{1}\{\mathbf{A}^1_{i,j} = a_0\} + \mathbb{1}\{\mathbf{M}_{i,j} \in \{F_1,F_2\}\}\\ &- p(\mathbf{M}^{\star\prime}_{i,j} = F_1|\mathbf{M}^\star_{i,j} \neq F_1)\mathbb{1}\{\mathbf{A}^1_{i,j} = a_0\} - \mathbb{1}\{\mathbf{M}^\star_{i,j} \in \{F_1,F_2\}\}\Big)\\ =&(1-1)C_6 + (0-0)\\ &+ C_3\Big(0 + 1 - 0 - 1\Big)\\ =&0\end{aligned}$$

So $c(s,a)$ is a subadditive function on $\mathcal{S} \times \mathcal{A}$.

Conditions Theorem 5.1: 1, 2, 3, → 4.

4. *"$q(k|s,a)$ is a superadditive function on $\mathcal{S} \times \mathcal{A}$ for all $k \in \mathcal{S}$ alternative: $\sum_{s'=0}^\infty p(s'|s,a)u(s')$ is a superadditive function on $\mathcal{S} \times \mathcal{A}$ for nonincreasing u." In dimension A^2.*

Let $\mathbf{A}^1$ be fixed, and suppose $s = \mathbf{M}$, $s+1 = \mathbf{M}^\star$, $a = (\mathbf{A}^1, a_0)$, $a^\star = (\mathbf{A}^1, a_2)$, so that $a \prec a^\star$.

For $A^2 = a_2$, the operation cycle ends, and the system returns to an as-good-as-new state, so $p^M(\mathbf{0}|s,a) = 1$. Then

$$\left[\sum_{s' \in \mathcal{S}^M} p^M(s'|s+1,a^\star)u(s') - \sum_{s' \in \mathcal{S}^M} p^M(s'|s+1,a)u(s')\right] - \left[\sum_{s' \in \mathcal{S}^M} p^M(s'|s,a^\star)u(s') - \sum_{s' \in \mathcal{S}^M} p^M(s'|s,a)u(s')\right]$$

$$
\begin{aligned}
&= \left[p^M(\mathbf{0}|s+1,a^\star)u(\mathbf{0}) - \sum_{s'\in\mathcal{S}^M} p^M(s'|s+1,a)u(s')\right] - \left[p^M(\mathbf{0}|s,a_1)u(\mathbf{0}) - \sum_{s'\in\mathcal{S}^M} p^M(s'|s,a)u(s')\right]\\
&= u(\mathbf{0}) - \sum_{s'\in\mathcal{S}^M} p^M(s'|s+1,a)u(s') - u(\mathbf{0}) + \sum_{s'\in\mathcal{S}^M} p^M(s'|s,a)u(s')\\
&= \sum_{s'\in\mathcal{S}^M} p^M(s'|s,a)u(s') - p^M(s'|s+1,a)u(s')\\
&= \sum_{s'\in\mathcal{S}^M} \Big(p^M(s'|s,a) - p^M(s'|s+1,a)\Big)u(s')\\
&= \sum_{\mathbf{M}'\in\mathcal{S}^M} \Big(p^M(\mathbf{M}'_{1,1}|\mathbf{M}_{1,1},\mathbf{A}^1_{1,1}) \times \ldots \times p^M(\mathbf{M}'_{m,n}|\mathbf{M}_{m,n},\mathbf{A}^1_{m,n})\\
&\quad - p^M(\mathbf{M}'_{1,1}|\mathbf{M}^\star_{1,1},\mathbf{A}^1_{1,1}) \times \ldots \times p^M(\mathbf{M}'_{m,n}|\mathbf{M}^\star_{m,n},\mathbf{A}^1_{m,n})\Big)u(\mathbf{M}')\\
&\overset{(1)}{=} \sum_{\mathbf{M}'\in\mathcal{S}^M} \Big(p^M(\mathbf{M}'_{1,1}|\mathbf{M}_{1,1},\mathbf{A}^1_{1,1}) \times \ldots \times p^M(\mathbf{M}'_{m,n}|\mathbf{M}_{m,n},\mathbf{A}^1_{m,n})\\
&\quad - p^M(\mathbf{M}'_{1,1}|\mathbf{M}^\star_{1,1},\mathbf{A}^1_{1,1}) \times \ldots \times p^M(\mathbf{M}'_{m,n}|\mathbf{M}_{m,n},\mathbf{A}^1_{m,n})\Big)u(\mathbf{M}')\\
&= \sum_{\mathbf{M}'\in\mathcal{S}^M} \Big(p^M(\mathbf{M}'_{1,1}|\mathbf{M}_{1,1},\mathbf{A}^1_{1,1}) \times \Omega - p^M(\mathbf{M}'_{1,1}|\mathbf{M}^\star_{1,1},\mathbf{A}^1_{1,1}) \times \Omega\Big)u(\mathbf{M}')\\
&= \Omega \sum_{\mathbf{M}'\in\mathcal{S}^M} \Big(p^M(\mathbf{M}'_{1,1}|\mathbf{M}_{1,1},\mathbf{A}^1_{1,1}) - p^M(\mathbf{M}'_{1,1}|\mathbf{M}^\star_{1,1},\mathbf{A}^1_{1,1})\Big)u(\mathbf{M}')\\
&= \Omega\Big[\sum_{\mathbf{M}'_{1,1}\preceq\mathbf{M}_{1,1}\prec\mathbf{M}^\star_{1,1}} \beta_1\Big(p(\mathbf{M}'_{1,1}|\mathbf{M}_{1,1},\mathbf{A}^1_{1,1}) - p(\mathbf{M}'_{1,1}|\mathbf{M}^\star_{1,1},\mathbf{A}^1_{1,1})\Big)u(\mathbf{M}')\\
&\quad + \sum_{\mathbf{M}'_{1,1}\preceq\mathbf{M}_{1,1}\prec\mathbf{M}^\star_{1,1}} \alpha_1\Big(p(\mathbf{M}'_{1,1}|\mathbf{M}_{1,1},\mathbf{A}^1_{1,1}) - p(\mathbf{M}'_{1,1}|\mathbf{M}^\star_{1,1},\mathbf{A}^1_{1,1})\Big)u(\mathbf{M}')\Big]\\
&\geq \Omega\Big[\sum_{\mathbf{M}'_{1,1}\preceq\mathbf{M}_{1,1}\prec\mathbf{M}^\star_{1,1}} \beta_1\Big(p(\mathbf{M}'_{1,1}|\mathbf{M}_{1,1},\mathbf{A}^1_{1,1}) - p(\mathbf{M}'_{1,1}|\mathbf{M}^\star_{1,1},\mathbf{A}^1_{1,1})\Big)u(\mathbf{M}')\\
&\quad + \sum_{\mathbf{M}'_{1,1}\preceq\mathbf{M}_{1,1}\prec\mathbf{M}^\star_{1,1}} \beta_1\Big(p(\mathbf{M}'_{1,1}|\mathbf{M}_{1,1},\mathbf{A}^1_{1,1}) - p(\mathbf{M}'_{1,1}|\mathbf{M}^\star_{1,1},\mathbf{A}^1_{1,1})\Big)u(\mathbf{M}')\Big]\\
&= \Omega\beta_1 \sum_{\mathbf{M}'_{1,1}\in\mathcal{S}} \Big(p(\mathbf{M}'_{1,1}|\mathbf{M}_{1,1},\mathbf{A}^1_{1,1}) - p(\mathbf{M}'_{1,1}|\mathbf{M}^\star_{1,1},\mathbf{A}^1_{1,1})\Big)u(\mathbf{M}')\\
&\geq 0
\end{aligned}
$$

The last inequality follows from Lemma 5.12, since $u(\mathbf{M}')$ is nonincreasing in $s = \mathbf{M}$ and $\sum_{\mathbf{M}'_{1,1}\in\mathcal{S}} p(\mathbf{M}'_{1,1}|\mathbf{M}^\star_{1,1}) \geq \sum_{\mathbf{M}'_{1,1}\in\mathcal{S}} p(\mathbf{M}'_{1,1}|\mathbf{M}_{1,1})$ as shown in the proofs from Theorems 5.4-5.8.

Lemma 5.12. *Let $\{x_i\}, \{x'_i\}$ be real-valued non-negative sequences satisfying $\sum_{j=k}^{\infty} x'_j \geq \sum_{j=k}^{\infty} x_j$ for all k, with equality holding for $k = 0$. Suppose $v_{j+1} \leq v_j$ for $j = 0, 1, \ldots$, then [46]*

$$\sum_{j=0}^{\infty} x'_j v_j \geq \sum_{j=0}^{\infty} x_j v_j.$$

□

5.6 Policy Iteration for Class-Ordered policies

The monotone class-ordered structure of the policy is used to effectively search through all policies and neglect certain suboptimal policies during policy iteration.

Puterman [46] describes a monotone policy iteration algorithm for totally ordered state space. We adapt this algorithm to make it applicable to the class-ordered state and action space, see the pseudocode in Algorithm 2. This algorithm is structured as follows:

1. *Initialization*: Randomly choose a nondecreasing policy π.
2. *Policy Evaluation*: Bellman's equations are solved via an iterative scheme to approximate the value of the current policy.
3. *Policy Improvement*: We improve the current policy by greedily selecting actions that minimizes the value function.

   We iterate over all states. When action a is assigned to state s, we also assign an action from the corresponding action class $\Phi(a)$ to all states within the same state class $\Theta(s)$.

   To guarantee the monotonicity of the new policy, we reduce the action space for the remaining states $s^\star$ in all state classes that strictly precede (succeed) the state class $\Theta(s)$, by removing all actions from $s^\star$'s action space that belong to the action classes strictly succeeding (preceding) the action class $\Phi(a)$.

The policy evaluation and policy improvement steps are repeated till the policy does no longer change. During the policy improvement step the policy can only decrease in cost, therefore the policy converges to the optimal policy.

Algorithm 2 Policy Iteration Algorithm for a Monotone Policy with Class-Ordered State and Action Space

```
Input: \gamma: discount factor; \theta: a small positive number.
Output: \pi^*: a deterministic optimal policy.

// Initialization
Randomly initialize a nondecreasing policy \pi in s

stable-policy = False
while stable-policy = False do
    // Policy Evaluation
    \pi_{old} = \pi
    \mathcal{A}'_s = \mathcal{A}_s
    \Delta = \theta + 0.1
    while \Delta > \theta do
        for s \in \mathcal{S}^M do
            v_{old} = v(s)
            v(s) = c(s, \pi_{old}(s)) + \gamma \sum_{s'} p(s'|s, \pi_{old}(s)) v(s')
            \Delta = \max\{\Delta, |v_{old} - v(s)|\}
        end for
    end while

    // Policy Improvement
    stable-policy = True
    visited-classes = emptylist
    for s \in \mathcal{S}^M do
        if \Theta(s) not in visited-classes then
            Add \Theta(s) to visited-classes
            \pi(s) = \arg\min_{a \in \mathcal{A}'_s} \{c(s,a) + \gamma \sum_{s'} p(s'|s,a) v(s')\}
            if \pi(s) != \pi_{old}(s) then
                stable-policy = False
            end if
            for \tilde{s} \in \Theta(s) and \tilde{s} \neq s do
                \pi(\tilde{s}) = \mathcal{A}'_{\tilde{s}} \cap \Phi(a)
            end for
            for classes \tilde{\Theta} not in visited-classes do
                if \tilde{\Theta} \prec \Theta(s) then
                    for s^\star \in \tilde{\Theta} do
                        \mathcal{A}'_{s^\star} = \mathcal{A}'_{s^\star} \setminus \bigcup_{\tilde{\Phi} \succ \Phi(\pi(s))} \tilde{\Phi}
                    end for
                else if \tilde{\Theta} \succ \Theta(s) then
                    for s^\star \in \tilde{\Theta} do
                        \mathcal{A}'_{s^\star} = \mathcal{A}'_{s^\star} \setminus \bigcup_{\tilde{\Phi} \prec \Phi(\pi(s))} \tilde{\Phi}
                    end for
                end if
            end for
        end if
    end for
end while
return \pi = \pi^*
```

6 Results

This section shows some results using the policy iteration for the single-unit model (Section 6.2) and the multi-unit model (Section 6.3).

6.1 Transition Function

The inputs for the transition probabilities are derived from the data analysis. The resulting transition matrix for the single-unit model is:

$$
p(s'|s,a_0) = \begin{array}{r|cccccccccc|}
 & (A_0,B_0,C_0,D_0,E_0) & (A_1,B_0,C_0,D_0,E_0) & (A_0,B_1,C_0,D_0,E_0) & (A_0,B_0,C_1,D_0,E_0) & (A_0,B_0,C_0,D_1,E_0) & (A_0,B_0,C_0,D_0,E_1) & \cdots & F_1 & F_2 \\
\hline
(A_0,B_0,C_0,D_0,E_0) & 0.977 & 0.004 & 0.004 & 0.008 & 0.004 & 0.002 & \cdots & 0 & 0 \\
(A_1,B_0,C_0,D_0,E_0) & 0.008 & 0.946 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\
(A_0,B_1,C_0,D_0,E_0) & 0.008 & 0 & 0.938 & 0 & 0 & 0 & \cdots & 0 & 0 \\
(A_0,B_0,C_1,D_0,E_0) & 0.007 & 0 & 0 & 0.945 & 0 & 0 & \cdots & 0 & 0 \\
(A_0,B_0,C_0,D_1,E_0) & 0.008 & 0 & 0 & 0 & 0.939 & 0 & \cdots & 0 & 0 \\
(A_0,B_0,C_0,D_0,E_1) & 0 & 0 & 0 & 0 & 0 & 0.958 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
F_1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 0 \\
F_2 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1
\end{array}
$$

The transition matrix for less state dimensions is made by neglecting the non-considered dimensions. For every individual dimension, this resulted in the following transition matrices:

- **State Dimension A: High Pressure**

$$
p(s'|s,a_0) = \begin{array}{r|ccccc|}
 & A_0 & A_1 & A_2 & F_1 & F_2 \\
\hline
A_0 & 0.9955 & 0.0044 & 0 & 0 & 0 \\
A_1 & 0.0047 & 0.990 & 0.0043 & 0 & 0 \\
A_2 & 0 & 0.0038 & 0.8821 & 0.0773 & 0.0366 \\
F_1 & 0 & 0 & 0 & 1 & 0 \\
F_2 & 0 & 0 & 0 & 0 & 1
\end{array}
$$

In the proof of Theorem 5.8 we assumed $p(A_0|A_0) \geq p(A_0|A_1)$ and $p(A_1|A_1) \geq p(A_1|A_2)$, both hold in the above transition matrix.

- **State Dimension B: CFI**

$$
p(s'|s,a_0) = \begin{array}{r|cccc|}
 & B_0 & B_1 & B_2 & F_1 \\
\hline
B_0 & 0.9934 & 0.01 & 0 & 0 \\
B_1 & 0.0060 & 0.9372 & 0.0058 & 0.0508 \\
B_2 & 0 & 0 & 0 & 1 \\
F_1 & 0 & 0 & 0 & 1
\end{array}
$$

In the proof of Theorem 5.5 we assumed $p(B_0|B_0) \geq p(B_0|B_1)$, this holds in the above transition matrix.

- **State Dimension C: Light Gas**

$$
p(s'|s,a_0) = \begin{array}{r|cccc|}
 & C_0 & C_1 & C_2 & F_2 \\
\hline
C_0 & 0.995 & 0.0043 & 0 & 0 \\
C_1 & 0.0042 & 0.9912 & 0.0044 & 0 \\
C_2 & 0 & 0.0042 & 0.9166 & 0.0790 \\
F_2 & 0 & 0 & 0 & 1
\end{array}
$$

In the proof of Theorem 5.7 we assumed $p(C_0|C_0) \geq p(C_0|C_1)$ and $p(C_1|C_1) \geq p(C_1|C_2)$, both hold according to the above transition matrix.

- **State Dimension D: Low Temperature**

$$p(s'|s,a_0) = \begin{array}{c|cccc} & D_0 & D_1 & D_2 & F_1 \\ \hline D_0 & 0.9935 & 0.0064 & 0 & 0 \\ D_1 & 0.0062 & 0.9369 & 0.0058 & 0.0509 \\ D_2 & 0 & 0 & 0 & 1 \\ F_1 & 0 & 0 & 0 & 1 \end{array}$$

In the proof of Theorem 5.6, we assumed $p(D_0|D_0) \geq p(D_0|D_1)$, this holds in the above transition matrix.

- **State Dimension E: Corrosion**

$$p(s'|s,a_0) = \begin{array}{c|ccccc} & E_0 & E_1 & E_2 & E_3 & F_2 \\ \hline E_0 & 0.9958 & 0.0041 & 0 & 0 & 0 \\ E_1 & 0 & 0.9957 & 0.02 & 0 & 0 \\ E_2 & 0 & 0 & 0.9 & 0.0042 & 0 \\ E_3 & 0 & 0 & 0 & 0.9210 & 0.0789 \\ F_2 & 0 & 0 & 0 & 0 & 1 \end{array}$$

6.2 Single-Unit Policies

One unit has $3 \times 3 \times 3 \times 3 \times 4 = 324$ deterioration states and two failure states, which makes a total of 326 possible states when all state dimensions are considered. We set $v(F_1) = v(F_2) = 1000$. There seems to be a bound for C_3, that once exceeded the policy returns stopping in all states, regardless of the value for $c_1(A_0, B_0, C_0, D_0, E_0)$. If we set the values for $v(F_1)$ and $v(F_2)$ a factor larger, the cost bound for C_3 also increases with the same factor. We use $\gamma = 0.7$ and $\gamma = 0.9$. The results are shown in Tables 10 - 13. In Tables 10 and 12, we considered all state dimensions and in Tables 11 and 13, we removed some dimensions. The run time for $\gamma = 0.9$ is longer than for $\gamma = 0.7$, because the policy evaluation step took more iterations. Action a_1 is chosen when the state reached or exceeded threshold τ_B or τ_D.

Dimensions	$c_1(\mathbf{0})$	C_3	τ_A	τ_B	τ_C	τ_D	τ_E	**Run Time 1**	**Run Time 0**
(A,B,C,D,E)	1	1	-	B_2	-	D_2	-	5.72 s	5.77 s
(A,B,C,D,E)	1	2.2	-	B_2	-	D_2	-	5.66 s	6.37 s
(A,B,C,D,E)	1	2.3	-	B_0	-	D_0	-	5.20 s	6.29 s
(A,B,C,D,E)	2	2.2	-	B_2	-	D_2	-	5.41 s	7.83 s
(A,B,C,D,E)	2.2	2.2	-	B_2	-	D_2	-	5.78 s	6.73 s
(A,B,C,D,E)	2	2.3	-	B_0	-	D_0	-	5.68 s	8.83 s
(A,B,C,D,E)	2.3	2.3	-	B_0	-	D_0	-	5.28 s	9.09 s
(A,B,C,D,E)	2.3	2.2	-	B_0	-	D_0	-	5.28 s	6.55 s

Table 10: Threshold for $\gamma = 0.7$. ‘Run Time 1’ is the run time using Algorithm 1, while ‘Run Time 0’ is the run time using standard policy iteration.

Dimensions	$c_1(\mathbf{0})$	C_3	τ_A	τ_B	τ_C	τ_D	τ_E	**Run Time 1**	**Run Time 0**
(A,B,C,D)	1	8.3	-	B_2	-	D_2	\	0.33 s	0.399 s
(A,B,C,D)	1	8.4	-	B_0	-	D_0	\	0.360 s	0.413 s
(A,B,C,E)	1	10.2	-	B_0	-	\	-	0.642 s	0.69 s
(A,B,C,E)	1	10.3	-	B_2	-	\	-	0.591 s	1.34 s
(A,B,C)	1	38.8	-	B_2	-	\	\	0.051 s	0.089 s
(A,B,C)	1	38.9	-	B_0	-	\	\	0.050 s	0.079 s
(B,C)	1	113.2	\	B_2	-	\	\	0.021 s	0.039 s
(B,C)	1	113.3	\	B_0	-	\	\	0.035 s	0.036 s

Table 11: Threshold for $\gamma = 0.7$. 'Run Time 1' is the run time using Algorithm 1, while 'Run Time 0' is the run time using standard policy iteration.

Dimensions	$c_1(\mathbf{0})$	C_3	τ_A	τ_B	τ_C	τ_D	τ_E	**Run Time 1**	**Run Time 0**
(A,B,C,D,E)	1	0.7	-	B_2	-	D_2	-	23.22 s	25.48 s
(A,B,C,D,E)	1	0.8	-	B_0	-	D_0	-	23.36 s	34.44 s
(A,B,C,D,E)	$c_1(s) = c_1 = 1$	0.8	-	B_0	-	D_0	-	29.58 s	32.4 s
(A,B,C,D,E)	$c_1(s) = c_1 = 1$	0.7	-	B_2	-	D_2	-	28.23 s	31.7 s
(A,B,C,D,E)	$c_1(s) = c_1 = 0.7$	0.7	-	B_2	-	D_2	-	28.11 s	31.5 s

Table 12: Threshold for $\gamma = 0.9$. 'Run Time 1' is the run time using Algorithm 1, while 'Run Time 0' is the run time using standard policy iteration.

Dimensions	$c_1(\mathbf{0})$	C_3	τ_A	τ_B	τ_C	τ_D	τ_E	**Run Time 1**	**Run Time 0**
(A,B,C,D)	1	2.7	-	B_2	-	D_2	\	0.965 s	1.10 s
(A,B,C,D)	1	2.8	-	B_0	-	D_0	\	0.907 s	0.921 s
(A,B,C,E)	1	3.4	-	B_0	-	\	-	0.642 s	0.69 s
(A,B,C,E)	1	3.5	-	B_2	-	\	-	0.591 s	1.34 s
(A,B,C)	1	12.9	-	B_2	-	\	\	0.196 s	0.115 s
(A,B,C)	1	13	-	B_0	-	\	\	0.187 s	0.116 s
(B,C)	1	37.7	\	B_2	-	\	\	0.044 s	0.021 s
(B,C)	1	37.8	\	B_0	-	\	\	0.031 s	0.024 s

Table 13: Threshold for $\gamma = 0.9$. 'Run Time 1' is the run time using Algorithm 1, while 'Run Time 0' is the run time using standard policy iteration.

6.3 Multi-Unit Policies

Recall that m is the number of flomels, n is the number of centrifuges in a single flomel. Location neighbouring units have more risk than process neighbouring units, so $d_P \leq d_L$ and these are estimated to be $d_L = 0.10$, $d_P = 0.08$. We use $\gamma = 0.9$.

We try different cost parameters. For state dimensions B and C, the results for 3 units are shown in Table 14. For dimension B, the results are shown for 3 units in Table 15, for 4 units in Table 16, and for 5 units in Table 17. We observed that action a_1 is chosen for a unit when it reaches or succeeds the state as shown in the table, but for $A^2 = a_2$, action a_1 is not chosen for any units. All solutions were found within two iterations.

$c_1(\mathbf{0})$	C_3	C_6	$\mathbf{A}^1_{i,j} = a_1$	$A^2 = a_2$	**Run Time 2**	**Run Time 0**
1,..,4	16	1,...,16	$\mathbf{M}_{i,j} \succeq (B_2, C_0)$	$\mathbf{M} \succeq \Theta((F_1, \mathbf{0}, \mathbf{0}))$	88.2 s	237.1 s
1	17	1,...,17	$\mathbf{M}_{i,j} \succeq (B_1, C_0)$	$\mathbf{M} \succeq \Theta((F_1, \mathbf{0}, \mathbf{0}))$	90.1 s	238.0 s
5,..,8	16	1,..,16	$\mathbf{M}_{i,j} \succeq (B_2, C_0)$	$\mathbf{M} \succeq \Theta((F_1, (B_2, C_0)), \mathbf{0}))$	103.3 s	295.4 s
9,...,16	16	1,...,16	$\mathbf{M}_{i,j} \succeq (B_2, C_0)$	$\mathbf{M} \succeq \Theta((F_1, F_1, \mathbf{0}))$	74,9 s	239,6 s
1	1	16	$\mathbf{M}_{i,j} \succeq (B_2, C_0)$	$\mathbf{M} \succeq \Theta((F_1, F_1, \mathbf{0}))$	69.2 s	219.0 s
1	2,3	16	$\mathbf{M}_{i,j} \succeq (B_2, C_0)$	$\mathbf{M} \succeq \Theta((F_1, (B_2, C_0), \mathbf{0}))$	65.4 s	228.8 s
1	4,..,15	16	$\mathbf{M}_{i,j} \succeq (B_2, C_0)$	$\mathbf{M} \succeq \Theta(F_1, \mathbf{0}, \mathbf{0})$	69.6 s	219.3 s
5	8	16	$\mathbf{M}_{i,j} \succeq (B_2, C_0)$	$\mathbf{M} \succeq \Theta((F_1, F_1, \mathbf{0}))$	69.9 s	226.4 s
1	16	17	$\mathbf{M}_{i,j} \succeq (B_2, C_0)$	$\mathbf{M} \succeq \Theta(F_1, \mathbf{0}, \mathbf{0})$	68.7 s	223.8 s
2,3,4	16	17	$\mathbf{M}_{i,j} \succeq (B_2, C_0)$	$\mathbf{M} \succeq \Theta((F_1, \mathbf{0}, \mathbf{0}))$	63.2 s	211.9 s
5,...,8	16	17	$\mathbf{M}_{i,j} \succeq (B_2, C_0)$	$\mathbf{M} \succeq \Theta((F_1, (B_2, C_0), \mathbf{0}))$	103.3 s	295.4 s
9,10	16	17	$\mathbf{M}_{i,j} \succeq (B_2, C_0)$	$\mathbf{M} \succeq \Theta((F_1, \mathbf{0}, \mathbf{0}))$	63.7 s	210.3 s

Table 14: State dimensions (B, C),$\gamma = 0.9$, $m = 3$, $n = 1$, Total 3 units. 'Run Time 2' is the run time using Algorithm 2, while 'Run Time 0' is the run time using standard policy iteration

$c_1(\mathbf{0})$	C_3	C_6	$\mathbf{A}^1_{i,j} = a_1$	$A^2 = a_2$	**Run Time 2**
1	16	16	$\mathbf{M}_{i,j} \succeq B_2$	$\mathbf{M} \succeq \Theta((F_1, B_0, B_0))$ $\vee$ $\mathbf{M} \succeq \Theta((B_1, B_1, B_0))$	6.61 s
1	17	1,...,17	$\mathbf{M}_{i,j} \succeq B_1$	$\mathbf{M} \succeq \Theta((B_1, B_1, B_0))$ $\vee$ $\mathbf{M} \succeq \Theta((F_1, B_0, B_0))$	3.06 s
1	1	16,17	$\mathbf{M}_{i,j} \succeq B_2$	$\mathbf{M} \succeq \Theta((B_2, B_2, B_0))$	3.10 s

Table 15: State dimension B,$\gamma = 0.9$, $m = 3$, $n = 1$, Total 3 units. 'Run Time 2' is the run time using Algorithm 2.

$c_1(\mathbf{0})$	C_3	C_6	$\mathbf{A}^1_{i,j} = a_1$	$A^2 = a_2$	**Run Time 2**	**Run Time 0**
0	16	16	$\mathbf{M}_{i,j} \succeq B_0$	$\mathbf{M} \succeq \Theta((F_2, B_0, B_0, B_0))$	14.65 s	310.0s
1,...,4	16	16	$\mathbf{M}_{i,j} \succeq B_2$	$\mathbf{M} \succeq \Theta((F_2, B_0, B_0, B_0))$ $\vee$ $\mathbf{M} \succeq \Theta((B_2, B_2, B_1, B_0))$	110.2 s	332.7 s
5,...,16	16	16	$\mathbf{M}_{i,j} \succeq B_2$	$\mathbf{M} \succeq \Theta((F_2, B_2, B_0, B_0))$ $\vee$ $\mathbf{M} \succeq \Theta((B_2, B_2, B_1, B_0))$	152.1 s	339.8 s
1	16	1,...,10	$\mathbf{M}_{i,j} \succeq B_2$	$\mathbf{M} \succeq \Theta((B_1, B_1, B_1, B_0))$ $\vee$ $\mathbf{M} \succeq \Theta((F_1, B_0, B_0, B_0))$	111.1 s	455.4 s

Table 16: State dimension B,$\gamma = 0.9$, $m = 4$, $n = 1$, Total 4 units. 'Run Time 1' is the run time using Algorithm 1, while 'Run Time 0' is the run time using standard policy iteration

$c_1(\mathbf{0})$	C_3	C_6	$\mathbf{A}^1_{i,j} = a_1$	$A^2 = a_2$	**Run Time 2**
16	16	1	$\mathbf{M}_{i,j} \succeq B_2$	$\mathbf{M} \succeq \Theta((B_2, B_2, B_2, B_0, B_0))$ $\vee$ $\mathbf{M} \succeq \Theta((F_1, B_2, B_1, B_0, B_0))$	4534.2 s

Table 17: State dimension B,$\gamma = 0.9$, $m = 5$, $n = 1$, Total 5 units. 'Run Time 2' is the run time using Algorithm 2.

7 Discussion

For the *single-unit model*, we observed that the resulting policy is influenced by thresholds in the cost components. Specifically, when the penalty C_3 is below a certain 'penalty' threshold, it is optimal to stop the unit if state-dimension B has reached or succeeded B_2 or if state-dimension D has reached or succeeded D_2. If the penalty C_3 is above the threshold, then the stopping action a_1 is already chosen in the as-good-as-new-state $(A_0, B_0, C_0, D_0, E_0)$, regardless of the reward values $c_1(s)$.

The dimension thresholds in this model appear to behave independently. Whether, all five dimensions or less dimensions were considered, the decision to choose the stopping action was not affected by the combined state of the other dimensions. This independence may suggest that threshold for each dimension can be determined separately.

A lower discount factor γ, which puts less emphasis on future costs (and more on immediate costs), allows a higher penalty C_3 value before the stopping action is chosen in a state.

For the *multi-unit model*, we considered only some of the state dimensions to reduce the run time. We specifically focused on dimension B, which ends in a crash, and dimension C, which ends in a run down. In reality, one might expect that is beneficial to stop and replace a sufficiently deteriorated (operating) centrifuge, when a refurbishment is planned. However, due to Assumption 5.5 that states that all centrifuges return to an as-good-as-new state after refurbishment, the model favoured to continue operation for single units, when the refurbishment action a_2 is chosen. This is because the unit then also returns to the as-good-as-new state with no additional replacement cost.

We also observed that a higher refurbishment cost C_6 does not lead to choosing the refurbishment action a_2 at a worse system state. Instead the action tends to be optimal in a better state. This arises because the total refurbishment costs are completely dependent on the number of operating units and there is no additional fixed component charged for every refurbishment .

Extending the single to a multi-unit model, resulted in an exponential increase in the size of the state and action space. Implementing the monotone policy structure in the policy iteration reduced the run times, and its impact was more noticeable in the multi-unit model compared to the single-unit model. Nevertheless, the run times still grow exponentially with each additional state dimension, which limited the accessibility of testing the multi-unit model.

In reality, there exists a, so-called, action-delay for refurbishment action a_2. The refurbishment cannot be immediately executed and is executed at a later decision epoch. Including the delay in the model requires tracking the previous actions within the state, which increases the size of the state. Therefore, we chose to neglect this delay. If the delay would be implemented, the refurbishment action would likely be chosen at an earlier state.

Deterioration patterns in centrifuges can vary from hours to years (e.g. corrosion). But a unit's state can become critical within seconds. Therefore, decisions must be made on the order of seconds and days. Given that our decision epochs are set to 15-second intervals, the probability of the system remaining in the same state is close to one. This often leads to a policy where units are stopped only in relatively very deteriorated conditions. However, in practice, operators cannot monitor every centrifuge every 15 seconds due to the size of the system. Therefore, longer intervals (like hourly decision epochs) may be more feasible for implementation. Longer intervals affect the transition probabilities since more state transitions are possible. Then it would be possible to reach a crash from a currently declared 'safe' state. The resulting policy would therefore, have a lower threshold, causing units to be stopped earlier.

We made several assumptions to simplify the problem and these provide opportunities for further research. First of all, the data analysis was not the prime focus of our research, hence assumptions were made about the relation between the observations and the true underlying state.

However, a more in-depth study would be needed to rigorously substantiate this relation.

We suggest evaluating additional time series and considering other factors, such as the amount of run ups that a centrifuge has experienced. More questions could be answered in the data analysis, for example *Are the frequency or amplitude from the seasonal component correlated?* or *Are the power and CFI-indicator correlated?*. Executing the decomposition approach for a single time series required several hours. Instead machine learning techniques could be employed to detect patterns more efficiently.

Centrifuges experience different effects based on their location in the system. Some stages are more susceptible to light gas, others more to CFI. This structural dependency can be implemented by assigning different transition functions to units based on their location.

8 Conclusion

Coming back to our research questions stated in Section 2.4:

1. *What kind of model can be used to determine a predictive maintenance policy?*
   We have formulated a Markov Decision problem, becomes an optimal stopping problem when only two actions are considered. For a single unit system, we considered production rewards, and crash penalties, to allow for a trade-off between maximizing operating hours and preventing a crash.

2. *How can a centrifuge's remaining useful life distribution be described?*
   We modelled the deterioration of a centrifuge using a discrete-time, multi-dimensional absorbing Markov chain. Each dimension represents a different deterioration path. The failure states are defined as absorbing states.

3. *What structure has the optimal policy for a single machine?*
   We proved that the optimal policy has a monotone structure. Given that we have two actions, the monotone policy is equivalent to a threshold policy, where the state space is divided in two mutually exclusive groups and one action is taken only within one of these groups. Due to the five-dimensional state space, the threshold is a five-way threshold.

4. *How can find the optimal policy?*
   We employed policy iteration to determine the optimal policy. Here, we used the monotone structure and the partial ordering of the state space to efficiently eliminate certain non-optimal policies.

5. *Does the policy structure change when a system of centrifuges is considered?*
   In a system of centrifuges it is also of interest to decide when to execute a refurbishment. So the policy involves two types of actions: whether to stop an individual machine and whether to execute a refurbishment. Although, the optimal policy is shown to have a monotone structure, it is not a threshold policy due to the multi-dimensional action space. The decision to stop a unit, now also depends on whether a refurbishment action is taken.

Now we can answer our main research question:

> “ Can we determine a policy to prevent crashes based on real-time observations? ”

We have determined a predictive maintenance policy to prevent centrifuge crashes by formulating a Markov decision process and solving it using policy iteration. We have developed a policy framework for both single and multi-unit centrifuge systems that indicates the cost-optimal moment to stop a centrifuge or perform a refurbishment based on the current condition of the system.

References

[1] M. Abdel-Hameed. “Deterioration processes”. In: *Semi-Markov Models: Theory and Applications*. Ed. by Jacques Janssen. Boston, MA: Springer US, 1986, pp. 231–252. ISBN: 978-1-4899-0574-1. DOI: 10.1007/978-1-4899-0574-1_13. URL: https://doi.org/10.1007/978-1-4899-0574-1_13.

[2] J.J. Arts. *Maintenance modeling and optimization*. BETA publicatie : working papers. Technische Universiteit Eindhoven, Mar. 2017.

[3] T. Aven. “Optimal replacement under a minimal repair strategy—a general failure model”. In: *Advances in Applied Probability* 15.1 (Mar. 1981), pp. 198–211. DOI: 10.2307/1426990. URL: https://doi.org/10.2307/1426990.

[4] A. Baru and R. Johnson. *Three ways to estimate remaining useful life for predictive maintenance*. 2020. URL: https://www.mathworks.com/content/dam/mathworks/ebook/estimating-remaining-useful-life-ebook.pdf.

[5] R. Bellman. “A Markovian decision process”. In: *Indiana University mathematics journal* 6.4 (Jan. 1957), pp. 679–684. DOI: 10.1512/iumj.1957.6.56038. URL: https://doi.org/10.1512/iumj.1957.6.56038.

[6] P. J. Brockwell and R. A. Davis. *Introduction to Time Series and Forecasting*. Jan. 2002. DOI: 10.1007/b97391. URL: https://doi.org/10.1007/b97391.

[7] H Burkill. “Monotonic functions on partially ordered sets”. In: *Journal of Combinatorial Theory Series A* 37.3 (Nov. 1984), pp. 248–256. DOI: 10.1016/0097-3165(84)90049-9. URL: https://doi.org/10.1016/0097-3165(84)90049-9.

[8] J. H. Cha, M. Finkelstein, and G. Levitin. “On preventive maintenance of systems with lifetimes dependent on a random shock process”. In: *Reliability Engineering System Safety* 168 (2017). Maintenance Modelling, pp. 90–97. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2017.03.023. URL: https://www.sciencedirect.com/science/article/pii/S0951832016305920.

[9] D. Chen and K. S. Trivedi. “Optimization for condition-based maintenance with semi-Markov decision process”. In: *Reliability Engineering System Safety* 90.1 (2005), pp. 25–29. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2004.11.001. URL: https://www.sciencedirect.com/science/article/pii/S0951832004002601.

[10] S. Christensen and A. Irle. “The monotone case approach for the solution of certain multidimensional optimal stopping problems”. In: *Stochastic Processes and their Applications* 130.4 (Apr. 2020), pp. 1972–1993. DOI: 10.1016/j.spa.2019.06.009. URL: https://doi.org/10.1016/j.spa.2019.06.009.

[11] Enrichment Technology Company. *Our Gas Centrifuge Technology for uranium enrichment — ETC*. Feb. 2023. URL: https://enritec.com/technology/centrifuge-technology/.

[12] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Apr. 2002. DOI: 10.1017/cbo9780511809088. URL: https://doi.org/10.1017/cbo9780511809088.

[13] B. De Jonge and P. Scarf. “A review on maintenance optimization”. In: *European Journal of Operational Research* 285.3 (Sept. 2020), pp. 805–824. DOI: 10.1016/j.ejor.2019.09.047. URL: https://doi.org/10.1016/j.ejor.2019.09.047.

[14] B. De Saporta et al. “Optimal stopping for the predictive maintenance of a structure subject to corrosion”. In: *Proceedings of the Institution of Mechanical Engineers. Part O, Journal of risk and reliability/Proceedings of the Institution of Mechanical Engineers. Part O, Journal of risk and reliability.* 226.2 (Oct. 2011), pp. 169–181. DOI: 10.1177/1748006x11413681. URL: https://doi.org/10.1177/1748006x11413681.

[15] R. Dekker et al. *MAINTENANCE AND MARKOV DECISION MODELS*. Dec. 2007. URL: https://math.leidenuniv.nl/reports/files/2007-39.pdf.

[16] M. Finkelstein. “On the optimal degree of imperfect repair”. In: *Reliability Engineering System Safety* 138 (June 2015), pp. 54–58. DOI: 10.1016/j.ress.2015.01.010. URL: https://doi.org/10.1016/j.ress.2015.01.010.

[17] T. Gao et al. "Data-Driven Method for predicting remaining useful life of bearing based on Bayesian theory". In: *Sensors* 21.1 (Dec. 2020), p. 182. DOI: 10.3390/s21010182.

[18] G.G. P. Garcia et al. "Interpretable Policies and the Price of Interpretability in Hypertension Treatment Planning". In: *Manufacturing Service Operations Management* 26.1 (Jan. 2024), pp. 80–94. DOI: 10.1287/msom.2021.0373. URL: https://doi.org/10.1287/msom.2021.0373.

[19] M. K. Goel, P. Khanna, and J. Kishore. "Understanding survival analysis: Kaplan-Meier estimate". In: *International journal of Ayurveda research* 1.4 (Jan. 2010), p. 274. DOI: 10.4103/0974-7788.76794. URL: https://doi.org/10.4103/0974-7788.76794.

[20] C. M. Grinstead Jr. et al. *Grinstead and Snell's Introduction to Probability*. Tech. rep. July 2006. URL: https://math.dartmouth.edu/~prob/prob/prob.pdf.

[21] H. Hartmann et al. "Overview of Toxicity Data and Risk Assessment Methods for Evaluating the Chemical Effects of Depleted Uranium Compounds". In: *Hum. Ecol. Risk Assess.* 6 (Oct. 2000). DOI: 10.1080/10807030091124239.

[22] *Hashed-feature similarity model for estimating remaining useful life - MATLAB - MathWorks Benelux*. Retrieved March 9, 2024. 2023. URL: https://nl.mathworks.com/help/predmaint/ref/hashsimilaritymodel.html.

[23] T. Hiruta and Y. Umeda. "Model-Based Deterioration Estimation with Cyber Physical System". In: *International journal of automation technology* 14.6 (Nov. 2020), pp. 1005–1012. DOI: 10.20965/ijat.2020.p1005. URL: https://doi.org/10.20965/ijat.2020.p1005.

[24] Nils Lid Hjort, Bent Natvig, and Espen Funnemark. "The Association in Time of a Markov Process with Application to Multistate Reliability Theory". In: *Journal of Applied Probability* 22.2 (1985), pp. 473–479. ISSN: 00219002. URL: http://www.jstor.org/stable/3213792 (visited on 10/14/2024).

[25] L. A. Jimenez-Roa et al. "Deterioration Modeling of Sewer Pipes via Discrete-Time Markov Chains; A Large-Scale Case Study in the Netherlands". In: *Book of Extended Abstracts for the 32nd European Safety and Reliability Conference*. ESREL. Research Publishing Services, 2022, pp. 1299–1306. DOI: 10.3850/978-981-18-5183-4_r22-13-482-cd. URL: http://dx.doi.org/10.3850/978-981-18-5183-4_R22-13-482-cd.

[26] L. Kallenberg. *MARKOV DECISION PROCESSES*. Tech. rep. Oct. 2016. URL: https://www.math.leidenuniv.nl/~kallenberglcm/Lecture-notes-MDP.pdf.

[27] M. J. Kim. "Robust Control of Partially Observable Failing Systems". In: *Operations Research* 64.4 (Aug. 2016), pp. 999–1014. DOI: 10.1287/opre.2016.1495. URL: https://doi.org/10.1287/opre.2016.1495.

[28] D. P. Kroese et al. "Data Science and Machine Learning: Mathematical and statistical methods". In: 77 (Jan. 2019).

[29] J. Li, Dirk Schaefer, and Jelena Milisavljevic-Syed. "A Decision-Based Framework for predictive maintenance technique selection in Industry 4.0". In: *Procedia CIRP* 107 (Jan. 2022), pp. 77–82. DOI: 10.1016/j.procir.2022.04.013. URL: https://doi.org/10.1016/j.procir.2022.04.013.

[30] B. H. Lindqvist. "Monotone and associated Markov chains, with applications to reliability theory". In: *Journal of Applied Probability* 24.3 (Sept. 1987), pp. 679–695. DOI: 10.2307/3214099. URL: https://doi.org/10.2307/3214099.

[31] D. Lugtigheid, D. Banjevic, and A. K.S. Jardine. "System repairs: When to perform and what to do?" In: *Reliability Engineering System Safety* 93.4 (Apr. 2008), pp. 604–615. DOI: 10.1016/j.ress.2007.03.023. URL: https://doi.org/10.1016/j.ress.2007.03.023.

[32] V. Makis and X. Jiang. "Optimal replacement under partial observations". In: *Mathematics of Operations Research* 28.2 (May 2003), pp. 382–394. DOI: 10.1287/moor.28.2.382.14484. URL: https://doi.org/10.1287/moor.28.2.382.14484.

[33] L. March, Cambridge Group for the History of Population, and Social Structure. *Measuring seasonality*, pp. 3–42. URL: https://www.demogr.mpg.de/books/drm/003/3.pdf.

[34] *MODULE 4.0: GAS CENTRIFUGE*. July 2009. URL: https://www.nrc.gov/docs/ML1204/ML12045A055.pdf.

[35] F. S. Nowlan and H. F. Heap. *Reliability-Centered maintenance*. Tech. rep. Dec. 1978. DOI: 10.21236/ada066579. URL: https://doi.org/10.21236/ada066579.

[36] Ivanka Barzashka Ivan Oelrich. *Federation of American Scientists :: Engineering Considerations for Gas Centrifuges*. Sept. 2013. URL: https://programs.fas.org/ssp/nukes/fuelcycle/centrifuges/engineering.html.

[37] Ivanka Barzashka Ivan Oelrich. *Federation of American Scientists :: Enrichment Cascades*. Sept. 2013. URL: https://programs.fas.org/ssp/nukes/fuelcycle/centrifuges/cascades.html.

[38] S. Oh and Ö. Özer. "Characterizing the structure of optimal stopping policies". In: *Production and Operations Management* 25.11 (Nov. 2016), pp. 1820–1838. DOI: 10.1111/poms.12579. URL: https://doi.org/10.1111/poms.12579.

[39] Donald R. Olander. "The theory of uranium enrichment by the gas centrifuge". In: *Progress in Nuclear Energy* 8.1 (1981), pp. 1–33. ISSN: 0149-1970. DOI: https://doi.org/10.1016/0149-1970(81)90026-3. URL: https://www.sciencedirect.com/science/article/pii/0149197081900263.

[40] *Pairwise comparison-based similarity model for estimating remaining useful life - MATLAB - MathWorks Benelux*. Retrieved March 9, 2024. 2023. URL: https://nl.mathworks.com/help/predmaint/ref/pairwisesimilaritymodel.html.

[41] M. D. Pandey and X. X. Yuan. *A Comparison of Probabilistic Models of Deterioration for Life Cycle Management of Structures*. Feb. 2007, pp. 735–746. DOI: 10.1007/1-4020-4891-2\{_}62. URL: https://doi.org/10.1007/1-4020-4891-2_62.

[42] K. Papadaki and W. B. Powell. "Monotonicity in multidimensional Markov decision processes for the batch dispatch problem". In: *Operations research letters* 35.2 (Mar. 2007), pp. 267–272. DOI: 10.1016/j.orl.2006.03.013. URL: https://doi.org/10.1016/j.orl.2006.03.013.

[43] J. Pike. *Gas centrifuge uranium enrichment*. URL: https://www.globalsecurity.org/wmd/intro/u-centrifuge.htm.

[44] *Probabilistic failure-time model for estimating remaining useful life - MATLAB - MathWorks Benelux*. Retrieved March 9, 2024. 2023. URL: https://nl.mathworks.com/help/predmaint/ref/reliabilitysurvivalmodel.html.

[45] *Proportional hazard survival model for estimating remaining useful life - MATLAB - MathWorks Benelux*. Retrieved March 9, 2024. 2023. URL: https://nl.mathworks.com/help/predmaint/ref/covariatesurvivalmodel.html.

[46] Martin L Puterman. *Markov decision processes*. en. Wiley Series in Probability and Statistics. Nashville, TN: John Wiley & Sons, Feb. 2005.

[47] *Residual comparison-based similarity model for estimating remaining useful life - MATLAB - MathWorks Benelux*. Retrieved March 9, 2024. 2023. URL: https://nl.mathworks.com/help/predmaint/ref/residualsimilaritymodel.html.

[48] Sheldon M. Ross. *Stochastic processes*. 2nd ed. Wiley, Dec. 1982. URL: https://openlibrary.org/books/OL3488929M/Stochastic_processes.

[49] Rajesh Siraskar et al. "Reinforcement learning for predictive maintenance: a systematic technical review". In: *Artificial Intelligence Review* 56.11 (Mar. 2023), pp. 12885–12947. DOI: 10.1007/s10462-023-10468-6. URL: https://doi.org/10.1007/s10462-023-10468-6.

[50] Youri Soons et al. *Predicting Remaining Useful Life with Similarity-Based Priors*. Jan. 2020, pp. 483–495. DOI: 10.1007/978-3-030-44584-3\{_}38. URL: https://doi.org/10.1007/978-3-030-44584-3_38.

[51] *Urenco*. URL: https://www.urenco.com.

[52] H. E. Van Staden, L. Deprez, and R. Boute. "A dynamic "predict, then optimize" preventive maintenance approach using operational intervention data". In: *European Journal of Operational Research* 302.3 (Nov. 2022), pp. 1079–1096. DOI: `10.1016/j.ejor.2022.01.037`. URL: `https://doi.org/10.1016/j.ejor.2022.01.037`.

[53] P. J. Vlok et al. "Optimal Component Replacement Decisions Using Vibration Monitoring and the Proportional-Hazards Model". In: *The Journal of the Operational Research Society* 53.2 (2002), pp. 193–202. ISSN: 01605682, 14769360. URL: `http://www.jstor.org/stable/822993` (visited on 08/02/2024).

[54] Y. Ye et al. "Modeling for reliability optimization of system design and maintenance based on Markov chain theory". In: *Computers Chemical Engineering* 124 (May 2019), pp. 381–404. DOI: `10.1016/j.compchemeng.2019.02.016`. URL: `https://doi.org/10.1016/j.compchemeng.2019.02.016`.

[55] J. Zhang et al. "Optimal inspection-based preventive maintenance policy for three-state mechanical components under competing failure modes". In: *Reliability Engineering System Safety* 152 (Aug. 2016), pp. 95–103. DOI: `10.1016/j.ress.2016.02.007`. URL: `https://doi.org/10.1016/j.ress.2016.02.007`.

[56] J. Zhao et al. "Overview of Equipment Health State Estimation and Remaining Life Prediction Methods". In: *Machines* 10.6 (May 2022), p. 422. DOI: `10.3390/machines10060422`. URL: `https://doi.org/10.3390/machines10060422`.

[57] Y. Zhou et al. "An effective approach to reducing strategy space for maintenance optimisation of multistate series–parallel systems". In: *Reliability Engineering System Safety* 138 (June 2015), pp. 40–53. DOI: `10.1016/j.ress.2015.01.018`. URL: `https://doi.org/10.1016/j.ress.2015.01.018`.

Appendices

A Glossary

Cascade	System of centrifuges that operate together.
Centrifuge Monitoring System	Measures various performance characteristics on centrifuge- and cascade level.
Chain	A totally ordered set, i.e. all elements in a chain are comparable.
CMS	Centrifuge monitoring system.
Crash	A failure type in which the rotor is destroyed.
Feed	Material that enters the centrifuge.
Flomel	A group of centrifuges in the cascade that receive the same feed and are on the same stage.
CFI	Condensed Feed Impurities. One of the failure causes for a centrifuge.
Markov chain	Type of Markov process, so it satisfies the Markov Property.
Markov Property	Future behaviours of the system only depends on the current state of the model, and not its history.
DTMC	Discrete time Markov chain.
PdM	Predictive Maintenance.
Poset	Partially ordered set.
Product	Enriched material that leaves the centrifuge.
Survival analysis	A class of statistical approaches to estimate the time for an event to occur.
Refurbishment	Maintenance action performed on the complete system, during which centrifuges can be replaced.
Rotor	Cylindrical chamber that rotates at high speed in the centrifuge, in which material is enriched.
RUL	Remaining useful life.
Run Down	The rotor decelerates to a complete stop. If can occur unsolicited when the centrifuge's friction exceeds the torque of its motor, in which case it is one of the failure types.
Run up	The centrifuge is turned on from a standstill and the rotor accelerates in several steps to normal frequency.
Tails	Depleted material that leaves the centrifuge.
UF6	Uranium hexafluoride. The feed material.

B Notation Index

Chapter 2

A_i	Deterioration state for High Pressure-path with $i = 0, 1, 2$
B_i	Deterioration state for Condensing Feed Impurities-path with $i = 0, 1, 2$.
C_i	Deterioration state for Light Gas-path with $i = 0, 1, 2$.
C_t	Cyclical component from time series at time t with $t = 1, 2, \ldots$.
D_i	Deterioration state for Low Temperature-path with $i = 0, 1, 2$.
d	Window size of the moving average filter.
E_i	Deterioration state for Corrosion-path with $i = 0, 1, 2, 3$.
e_t	Residuals from decomposed time series at time t with $t = 1, 2, \ldots$.
F_1	Run Down state, absorbing failure state.
F_2	Crash state, absorbing failure state.
S_t	Seasonal component from time series at time t with $t = 1, 2, \ldots$.
T_t	Trend component from time series at time t with $t = 1, 2, \ldots$.
X_t	Time series with $t = 1, 2, \ldots$.

Chapter 5
Single-Unit Model

a_0	'continue'-action, there is no effect on the current state.
a_1	'stop'-action, the unit is stopped preemptive via a Run Down.
$\mathcal{A}$	Action space for the single-unit model.
$c(s, a)$	Cost function, returns the immediate cost when action a is taken in state s.
c_{crash}	Penalty cost for an unexpected crash.
$c_{\text{product}}(s, a)$	Reward for production (nonpositive costs).
C_3	Nonnegative cost parameter that represents the cost of a crash.
$h^*_\gamma(s)$	used for Assumption 5.3
$p(s'\|s, a)$	Probability to transition to state s' given the current state is s and action a is taken.
$p(s'\|s)$	Probability to transition to state s' given the current state is s and action a_0 is taken.
π^*	Optimal policy that minimizes the costs.
$q(k\|s, a)$	used for Condition 2 in 5.1.
$\mathcal{S}$	State space for the single-unit model, $\mathcal{S} = \mathcal{S}_A \times \mathcal{S}_B \times \mathcal{S}_C \times \mathcal{S}_D \times \mathcal{S}_E$.
$\mathcal{S}_A$	State space for dimension A.
$\mathcal{S}_B$	State space for dimension B.
$\mathcal{S}_C$	State space for dimension C.
$\mathcal{S}_D$	State space for dimension D.
$\mathcal{S}_E$	State space for dimension E.
τ_A	Threshold for dimension A, the stopping action a_1 becomes optimal once state exceeds the threshold.
τ_B	Threshold for dimension B.
τ_C	Threshold for dimension C.
τ_D	Threshold for dimension D.
τ_E	Threshold for dimension E.
$v^\pi_\gamma(s)$	Value function with γ-discounted expected costs.
Z	Nonnegative value assigned to the value function for the failure states.
$\preceq, \succeq$	Partial ordering operators.

Multi-Unit Model

a_0	'continue operating'-action, there is no effect on the current state
a_1	'stop'-action, the unit is stopped preemptive via a Run Down.
a_2	refurbishment is planned
a_3	No action is taken, this action is only available for non-operating units
$\mathcal{A}$	Action space for the multi-unit model.
α	Substitute variable.
β	Substitute variable.
$c(s,a)$	Cost function
$c_{\text{crash}}(s,a)$	Penalty cost for an unexpected crash
$c_{\text{non-operating}}(s,a)$	Cost for every non-operating unit
$c_{\text{product}}(s,a)$	Reward for production (nonpositive costs)
$c_{\text{replacement}}(s,a)$	Replacement cost of all non-operating centrifuges
C_3	Nonnegative cost parameter that represents the cost of a crash
C_6	The crew cost and the purchase cost of a new unit
D_i	Factor that increases the probability of transitioning to a worse state
d_L	Factor if crash occurred in a location-neighbouring flomel
d_P	Factor if crash occurred in a process-neighbouring flomel
m	Number of flomels
n	Number of centrifuges in one flomel
$N_{l,i}$	Set of location-neighbouring flomels of flomel i
$N_{P,i}$	Set of process-neighbouring flomels of flomel i
π^*	Optimal policy that minimizes the costs
$p^M(s'\|s,a)$	Transition function for the multi-unit system.
$p(\mathbf{M}'_{i,j}\|\mathbf{M}_{i,j})$	Transition function for a single unit, when action a_0 is chosen for a single unit. Same function as in the single-unit model.
Q_i	Factor that decreases the probability of transitioning to the same or a better state
$\mathcal{S}^M$	State space for the Multi-unit model
$\mathcal{S}$	State space for a single unit. Same as in the single-unit model.
$v^\pi_\gamma(s)$	Value function with γ-discounted expected costs
$\preceq, \succeq$	Partial ordering operators.

C Assumptions

- The initial state of the system is as-good-as-new.
- Centrifuges have an increasing failure rate.
- Centrifuges have a finite lifetime.
- Centrifuges are subject to deterioration.
- The centrifuge is monitored every certain period of time. The component's state is considered unobservable in between these times. The inspections are non-destructive.
- Observations directly relate to a centrifuge's true deterioration state.
- Changes in the centrifuge's state only depend on the current state and the action taken.
- The reward function and transition probabilities are not dependent on time, i.e. we have stationary rewards and transition probabilities.
- The rewards are bounded for all actions and states.
- The state- and action space are discrete and finite.
- Future rewards are discounted according to a discount factor γ, with $0 \leq \gamma < 1$.

- After refurbishments all centrifuges (replaced and non-replaced) are considered to be as-good-as-new.

D Imputation methods for missing values

Mean Imputation: fills in the missing values by the average of the whole column. This method fails to consider trends or seasonality, but is suitable when data is assumed to be randomly distributed.

Median Imputation: Replace missing values by the median of the entire column.

Last Imputation Carried Forward: Replaces missing values with the last known value. This works well for time series with a constant or rising trend, but distorts trends that are changing.

Next Observation Carried Forward: Replaces missing values with the next known value. This works well for time series with a constant or downwards trend.

Linear Interpolation: Estimates missing values by drawing a linear line between the two nearest known values. This Works well for linear trends, but fails to describe complex trends.

Spline Interpolation: Estimates missing values by fitting a curved line through the data points. This is computationally more expensive than linear interpolation.

E Remaining Useful Life (continued)

Data-driven techniques are classified in two classes, depending whether a probability distribution of the RUL must be obtained, or a point-estimate is sufficient. A probability distribution of RUL can be implemented in stochastic decision making, which is our interest [50].

E.A Similarity Models

Similarity Models are based on the hypothesis that if at some point a curve has evolved in the same way as other curves, it is likely to continue to do so, and therefore have a similar remaining useful life (RUL). Similarity models estimate the RUL of the test unit as the median statistic T of the lifetime span of the most similar units (in the training set) minus the current lifetime value t of the test unit; $T - t$. Different Similarity Models exist. Which model is suitable depends on the available data [4]:

- A **Hash Similarity Model** is useful when **large data sets** are available. This model transforms the historical deterioration path data for each member in the data set in a series of hashed-features. These features may be the mean, minimum, or maximum values for the data. The hashed features of the test unit can be computed and compared to the features of the data members [22].

- A **Pairwise Similarity Model** compares the deterioration profile of a test unit directly to the deterioration path histories for multiple similar units. The similarity of the test unit to the other units is a function of the distance between the deterioration profile and the units profile, computed using correlation or dynamic time warping. [40].

- A **Residual Similarity Model** is useful when the **deterioration dynamics** of multiple similar units are known. The historical data for each member in the data is fitted with a model. The deterioration data of the test unit is then used to compute one-step prediction errors, or residuals for the model with each data member. The error sizes show the similarity of the test unit to the members [47].

E.B Survival Models

Survival models are used when the only available data are the failure times of similar units. Two different Survival Models are distinguished:

- A **Reliability Survival Model** is used when the only available data is the failure times of multiple similar units. The coefficients of a probability distribution are estimated using the failure-time data. The survival model then uses the probability distribution of unit failure times to estimate the RUL [44].
- A **Covariate Survival Model** (also called a **Proportional Hazard Survival Model**) is used when next to life-time data one also has associated covariates. Associated covariates can be environmental or explanatory variables. The model coefficients are estimated using the collection of failure-time and associated covariates [45].

F Deterministic Deterioration Models

F.A Failure patterns

There are six different failure patterns associated with equipment failure: The Bathtub curve, Wear-out curve, Fatigue Wear curve, Initial Break-in period, Constant Failure rate, and Infant Mortality. See Figure 18 for the different shapes of the failure patterns, where the failure rate of equipment is plotted against time. For most patterns the failure rate varies over time. The shape of the pattern allows to identify whether a failure is an infant mortality, random, or wear-out failure. The first two curves have a well-defined wearout period. For these two curves, an age limit may be useful, but the effectiveness depends on the probability that a unit survives to that age [35].

The failure pattern of a centrifuge follows a Bathtub curve. This curve has three sections. The first section is the 'run-up period', where the failure rate is decreasing and failures are mostly caused by defective units. A unit remains most of its lifespan in the 'random failure period', which has a low constant failure rate since units are worn in. The third section is the 'wear-out period', where the failure rate increases again, as wear and mechanical fatigue have caused damage over time resulting in failures [35].

F.B Failure Rates

The failure rate is defined as the frequency with which a unit fails:

Definition F.1. Failure rate $h(t)$ is the limit of the probability that a failure occurs per unit time interval Δt, given that no failure has occurred before time t:

$$h(t) = \lim \frac{\mathbb{P}(t < T \leq t + \Delta t | T > t)}{\Delta T} = \frac{\lim_{\Delta t \to 0} \{F(t + \Delta t) - F(t)\}/\Delta t}{R(t)} = \frac{f(t)}{R(t)},$$

with $R(t)$ the probability that the unit survives beyond time t, and $f(t)$ the probability density function of the lifetime.

We have

$$R(t) = \int_t^\infty f(s)ds = 1 - \int_0^t f(s)ds.$$

The lifetime is often assumed to be of a distribution (Weibull, exponential, or gamma), or a given family of distributions (increasing failure rate, or increasing failure rate average). The failure rates are estimated based on data collected on the failure times and verified by hypothesis testing. In practice there is often not enough data available, or the validity of the distribution is questionable due to the sometimes unfounded assumptions. The above difficulties can be avoided by looking into the way a failure occurs and use this to determine the form of its distribution function [1].

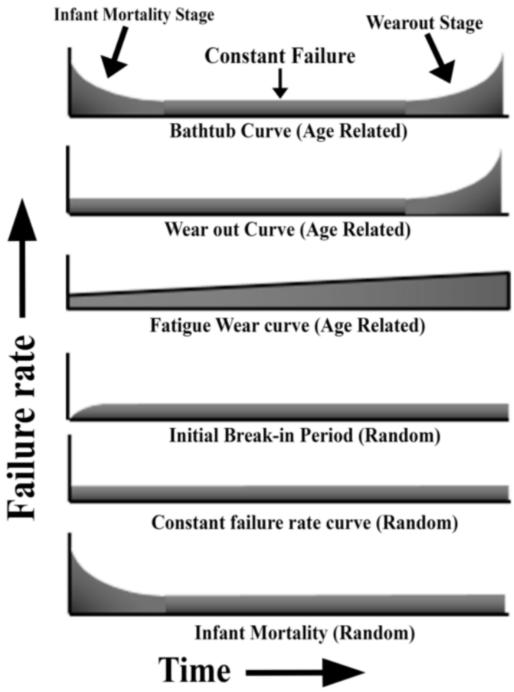


Figure 18: The six failure patterns as defined by Nolan and Heap [35]

The Weibull distribution is the most used lifetime distribution in reliability analysis. This distribution is described by

$$f(\lambda, \beta, t) = \lambda\beta(\lambda t)^{\beta-1} \exp^{-(\lambda t)^{\beta}}, \quad \lambda, \beta, t > 0,$$

with scale parameter λ, which indicates the unit's potential of failure, and shape parameter β. Three cases are described for β [54]:

- $\beta = 1$: Weibull reduces to an exponential distribution $f(t) = \lambda e^{-\lambda t}$, so $R(t) = e^{-\lambda t}$, and the failure rate $h(t) = \lambda$ is constant;
- $\beta > 1$: $R(t) = \exp^{-(\lambda t)^{\beta}}$, and the failure rate $h(t) = \beta\lambda^{\beta} t^{\beta-1}$ increases with time;
- $\beta < 1$: $R(t)$ and $h(t)$ are the same as for $\beta > 1$, but now the failure rate decreases with time.

G Proofs of theorems

Theorem 4.1

Proof. A characteristic from an absorbing Markov chain is that it is possible to reach an absorbing state from any transient state s_j. Let m_j denote the minimum number of steps needed to reach an absorbing state from state s_j. The probability to not reach an absorbing state in m_j steps is given by p_j, with $0 \leq p_j < 1$. Let $m = \min_j m_j$, and $p = \max_j p_j$. Then the probability of no absorption in m steps is less than or equal to p. No absorption in $2m$ steps is less than or equal to p^2. So the probability of no absorption within nm steps is p^n. Since $p < 1$, the probabilities monotone decrease to zero, hence $\lim_{n\to\infty} \mathbf{Q}^n = 0$. □

Theorem 4.2

Proof. First, we proof that there exists an inverse for matrix $\mathbf{I} - \mathbf{Q}$. For a square matrix the inverse exists when it's non-singular, i.e. $\mathbf{1}$ is not an eigenvalue of $\mathbf{Q}$. $\mathbf{x} = \mathbf{Q}\mathbf{x}$ Let $(\mathbf{I} - \mathbf{Q})\mathbf{x} = 0$, which equals $\mathbf{x} = \mathbf{Q}\mathbf{x}$. Iterating, we obtain $\mathbf{x} = \mathbf{Q}^n\mathbf{x}$ Since $\mathbf{Q}^n \to \mathbf{0}$, we have $\mathbf{Q}^n\mathbf{x} \to \mathbf{0}$, together with $\mathbf{x} = \mathbf{Q}^n\mathbf{x}$,we have $\mathbf{x} = 0$. So the inverse $(\mathbf{I} - \mathbf{Q})^{-1} = \mathbf{N}$ exists. Next, we can write

$$(\mathbf{I} - \mathbf{Q})(\mathbf{I} + \mathbf{Q} + \mathbf{Q}^2 + \ldots + \mathbf{Q}^n) = I - \mathbf{Q}^{n+1}$$

. Multiplying both sides by $(\mathbf{I} - \mathbf{Q})^{-1}$, gives

$$(\mathbf{I} - \mathbf{Q})^{-1}(\mathbf{I} - \mathbf{Q})(\mathbf{I} + \mathbf{Q} + \mathbf{Q}^2 + \ldots + \mathbf{Q}^n) = (\mathbf{I} - \mathbf{Q})^{-1}(I - \mathbf{Q}^{n+1})$$

$$(\mathbf{I} + \mathbf{Q} + \mathbf{Q}^2 + \ldots + \mathbf{Q}^n) = (\mathbf{I} - \mathbf{Q})^{-1}(I - \mathbf{Q}^{n+1})$$

Let $n \to \infty$, so that $\mathbf{Q}^{n+1} \to \mathbf{0}$ gives

$$(\mathbf{I} + \mathbf{Q} + \mathbf{Q}^2 + \ldots) = (\mathbf{I} - \mathbf{Q})^{-1} = \mathbf{N}.$$

Lastly, we proof that entry n_{ij} denotes the expected number of times the chain is in state s_j given starting in state s_i.

Let states s_i and s_j be two transient states. We introduce binary variable $X^{(k)}$ with $X^{(k)} = 1$ if the chain is in state s_j after k steps, $X^{(k)} = 0$ otherwise. For each k the binary variable depends on i and j. Then $\mathbb{P}(X^{(k)} = 1) = q_{ij}^{(k)}$, and $\mathbb{P}(X^{(k)} = 0) = 1 - q_{ij}^{(k)}$ for $k = 0, 1, 2, \ldots$, where $q_{ij}^{(k)}$ is the (i, j)-th entry of $\mathbf{Q}^k$. This gives $\mathbb{E}(X^{(k)}) = \mathbb{P}(X^{(k)} = 1)1 + \mathbb{P}(X^{(k)} = 0)0 = \mathbb{P}(X^{(k)} = 1) = q_{ij}^{(k)}$. Similarly,

$$\mathbb{E}(X^{(0)} + X^{(1)} + \ldots + X^{(n)}) = \mathbb{E}(X^{(0)}) + \mathbb{E}(X^{(1)}) + \ldots + \mathbb{E}(X^{(n)}) = q_{ij}^{(0)} + q_{ij}^{(1)} + \ldots + q_{ij}^{(n)}$$

Let $n \to \infty$, then

$$\mathbb{E}(X^{(0)} + X^{(1)} + \ldots + X^{(n)}) = \mathbb{E}(X^{(0)}) + \mathbb{E}(X^{(1)}) + \ldots = q_{ij}^{(0)} + q_{ij}^{(1)} + \ldots = n_{ij}$$

□