Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

Markus Bratland Kvammen

Assessing Maintenance Problems Using a Proof-of-concept DES+SD Modeling and Simulation Tool

Master's thesis in Engineering and ICT
Supervisor: Per Schjølberg
Co-supervisor: Antoine Rauzy, Jon Martin Fordal
June 2021

NTNU
Norwegian University of
Science and Technology

Markus Bratland Kvammen

Assessing Maintenance Problems Using a Proof-of-concept DES+SD Modeling and Simulation Tool

Master's thesis in Engineering and ICT
Supervisor: Per Schjølberg
Co-supervisor: Antoine Rauzy, Jon Martin Fordal
June 2021

Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

NTNU
Norwegian University of
Science and Technology

Abstract

This project aims to study how modeling and simulation can help to understand industrial maintenance problems better. The purpose is to help managers understand their problems better, the root cause, and potential solutions. Despite substantial development of maintenance managing models and optimization tools, there is a frequently emphasized gap between theory and practice. To address this, we developed a proof-of-concept modeling and simulation tool using a combined framework of discrete event simulation (DES) and system dynamics (SD). Further on, we performed five different experiments to study features of the maintenance process.

The first experiment serves as a base case for the following experiments. The experiment simulates the accumulation of preventive maintenance (PM) backlog due to an imbalance between the capacity to carry out maintenance and the PM programs. In experiments two and three, we study two different ways of reducing the backlog. First, we increase the PM interval, and secondly, increase the capacity to perform maintenance. Both approaches reduced the PM backlog. The latter, however, also reduces the actual PM interval, which increases the number of PM activities performed.

Further on, in experiment four, we study how the system responds to an unexpected event. The results show that a highly optimized maintenance process might be extremely fragile to unexpected events and delays. Lastly, in experiment five, we investigate how inaccurate prioritization of PM activities affect the maintenance process. The experimental results show that inaccurate prioritization might lead to substantial risk exposure. The experiment also shows how the PM backlog appears to be in equilibrium and indicates a well-functioning PM program without any PM activities performed. In combination with the data management policy, the corrective reality leads to rescheduling of PM activities until the item's failure.

Future research should continue the development of a multi-perspective modeling framework and methodology to understand industrial problems better. For the continued development of a modeling framework, we recommend exploring the combination of DES and SD. The methodology should emphasize the creation of metaphor models and exploration of the dynamics of variables. We envision models created with such a modeling framework to serve as both a learning laboratory and as a communication tool among stakeholders.

Furthermore, for addressing the challenges of managing maintenance, we recommend future research to expand the system boundary, including, for instance, production, spare part policy, item criticality, and soft variables such as competence and pressure. Moreover, study the dynamics of variables considered static in this preliminary—for instance, PM prioritizing accuracy along with the PM backlog, data quality, and scheduling policy. Additionally, further development and research would benefit from collaboration with industrial companies. A collaboration would improve the development to include the factors affecting their specific problems and to ensure the usefulness of such modeling framework and tools to the industry.

Sammendrag

Dette prosjektet har som formål å studere hvordan modellering og simulering kan bidra til å bedre forstå industrielle vedlikeholdsproblemer. Hensikten er å hjelpe ledere til å forstå deres problemer bedre, finne kilden til problemene, og potensielle løsninger. Til tross for en betydelig utvikling av vedlikeholdsstyringskonsepter og optimaliseringsverktøy, er det er ofte vektlagt skille mellom teori og praksis. For å adressere dette har vi utviklet et modellering og simuleringsverktøy ved bruk av konsepter fra diskret hendelsessimulering (DES) og systemdynamikk (SD). Videre har vi gjennomført fem eksperimenter for å studere egenskaper ved vedlikeholdsprosessen.

Det første eksperimentet danner basisgrunnlaget for de følgende eksperimentene. Eksperimentet simulerer en akkumulering av et preventivt vedlikehold (PM) etterslep som følge av en ubalanse mellom kapasiteten til å utføre vedlikehold og PM programmene. I eksperiment to og tre studerer vi to ulike måter å redusere etterslepet. Først øker vi PM intervallet og deretter øker vi kapasiteten til å utføre vedlikehold. Begge tiltakene reduserer PM etterslepet. På en annen side reduserer det siste også det faktiske PM intervallet, som igjen øker mengden PM aktiviteter som blir gjennomført.

I eksperiment fire studerer vi hvordan systemet responderer på en uventet hendelse. Resultatet viser at en høyst optimalisert vedlikeholdsprosess kan være svært sårbar for uventede hendelser og forsinkelser. Til slutt, i eksperiment fem, undersøker vi hvordan unøyaktig prioritering av PM-aktiviteter påvirker vedlikeholdsprosessen. Resultatene viser at unøyaktig prioritering kan føre til en betydelig risiko. Eksperimentet viser hvordan PM ettersleppet tilsynelatende er stabilt, og indikerer et velfungerende PM program, men uten at det blir gjennomført PM-aktiviteter. Prinsippet for håndtering av data i kombinasjon med en korrektiv virkelighet, fører til gjentatt planlegging og forskyvning av PM aktiviteter frem til enheten svikter.

Videre forsking burde fortsette utviklingen av et flerperspektiv-modelleringsrammeverk og -metodikk for å forstå industrielle problemer bedre. For videre utvikling av dette anbefaler vi å utforske kombinasjonen DES og SD. Metodikken burde legge vekt på utvikling av metaformodeller og utforsking av dynamikken til variabler. Vi forestiller oss at modeller utviklet i et slikt modelleringsrammeverk kan bidra som både et læringsverktøy, og som et kommunikasjonsverktøy mellom interessenter.

For videre utforsking av utfordringer med vedlikeholdsstyring anbefaler vi å utvide grensene til systemet, og inkludere for eksempel produksjon, reservedeler, kritikalitet, og myke variabler, som kompetanse og press. Dessuten, studere dynamikken til variablene som i denne innledende forskingen er betraktet som statisk. For eksempel studere nøyaktigheten i prioritering av PM i sammenheng med PM etterslep, datakvalitet, og policy for planlegging. Et samarbeid med industrielle virksomheter vil være til fordel ved videre forsking og utvikling. Et slikt samarbeid vil forbedre utviklingen ved å inkludere faktorer som påvirker deres problemer, og bidra til å sørge for nytteverdien av et slikt verktøy og modelleringsrammeverk for industrien.

Acknowledgements

I would like to mark my gratitude to my supervisors for introducing me to the adventurous life of academic research. Their help, inspiration, and guidance during this project is very much appreciated. A special thanks to Ass. Professor Per Schjølberg, for putting me in contact with Equinor, and to Professor Antoine Rauzy, for numerous inspiring discussions. I would also mark my gratitude to the industrial contact, Nils Martin Rugsveen, for the valuable insight he has provided into the maintenance operation at Equinor.

All good things must come to an end. However, the journey of this project has merely begun. The master's thesis serves as preliminary work for an upcoming Ph.D.

I look forward to continuing the research project and collaboration with my supervisors and industrial contact.

Markus B. Kvammen

Contents

1 Introduction

In the past decades, there has been a substantial change in the field of maintenance. In the 1960s and 1970s, academics started to use mathematics in order to optimize maintenance. These optimization formulas, however, suffered from being too simplified, leading to a gap between theory and practice, between academics and industry (Pintelon and Parodi-Herz, 2008). Over the years, this gap has decreased, and new optimization methods have been developed, such as simulation. Furthermore, in the past decades, the role of technology in the maintenance process has increased. Computerized maintenance management software, for instance, allows to keep track of maintenance records, extract data to give input to statistical analysis, plan and schedule maintenance jobs, and keep track of all the items in the fleet. While sensors, on the other hand, enable the measurement of features such as vibration and temperature, connecting the physical and the digital world, all in an attempt to monitor the condition of an item, collect data and raise alarms.

In the past years, there has been an increasing attempt to use machine learning and Big Data techniques to connect the various sources of maintenance data with the sensor data available to predict the future development of the state of the items. The goal of these attempts is to enable the scheduling maintenance based on a predicted remaining useful lifetime of an item, called predictive maintenance (Pedersen, Vatn, and Jørgensen, 2020). Although these rapidly growing optimization tools and technological developments are assumed to improve the performance of a maintenance organization (Bokrantz et al., 2017), researchers argue that there is a need for a holistic perspective on maintenance for successful development and to close the gap between theory and practice (Bengtsson and Salonen, 2009). The lack of a holistic perspective, researchers argue, is why although a few companies have huge success with the implementation of innovations and tools, most efforts fail to produce significant results. The unsuccessful implementation is a systemic problem created by the interactions between the existing system of items, tools, workers, and managers and the innovations and tools (Repenning and Sterman, 2003).

Researchers have proposed modeling and simulation as a tool to capture the complex nature of the maintenance process. This could enable companies to trace the effect of maintenance policies and close the gap between theory and practice (Alrabghi and Tiwari, 2016). Although simulations have been applied to model the maintenance process, these are often limited to isolated parts and do not consider the holistic view of the process (Alabdulkarim, Ball, and Tiwari, 2013). System dynamics (SD) is an approach suitable for assessing and improving the performance of complex organizations, as well as explaining their behavior in a holistic, long-term perspective (Sterman, 2000). SD has shown great results in terms of supporting managers to improve the maintenance organization (Carroll, Sterman, and Marcus, 1998). While SD models help capture the complex behavior of an organization at a holistic level, the approach is unsuitable for investigating problems requiring a high level of detail (Linnéusson, 2018). A simulation technique suitable for capturing industrial systems' details, such as the maintenance process, is discrete event simulation (DES). DES is a technique where a system is represented as entities flowing through a network of queues and activities in discrete points in time (Robinson and Brailsford, 2014). Although DES is widely used in operation research due to its flexibility to model a wide variety of systems (Robinson and Brailsford, 2014), researchers have criticized DES studies for neglecting the feedback behavior, which helps explain behaviors of the system (Günal and Pidd, 2010). Insight into the behavior of the system is necessary for strategic development and is supported by simulation of SD models (Warren, 2005). Researchers have proposed to combine SD with DES to capture both the holistic and detailed perspective in short- and long-terms (Linnéusson, 2018). Although this combination is believed to have a vast potential within operations management (Morgan, Howick, and Belton, 2017), the research of this combined approach within the maintenance field is limited, and very few works demonstrate simulation models and results (Linnéusson, Ng, and Aslam, 2018). Thus, there is unexplored potential in combining DES and SD to develop a better understanding of complex maintenance problems.

1.1 Maintenance at Equinor

Equinor is the largest energy company in Norway and employs more than 21,000 globally. They focus their work on both oil and gas and renewable energy sources such as offshore wind. The petroleum safety authority in Norway (Ptil) has several times reported substantial backlogs of both preventive and corrective maintenance in their facilities (see, e.g., Thorsen, Leto, and Jåsund (2020) and Jåsund et al. (2014)). In their investigation of Equinor's process facility at Mongstad in 2020, Ptil discovered 42 000 hours of PM backlog and 255 000 hours of CM backlog. These numbers are approximately the same amount as Ptil reported from the same facility in 2019. Equinor's objective, however, is to have less than 1000 hours of PM backlog and less than 50 - 80 000 hours of CM backlog (Thorsen, Leto, and Jåsund, 2020). Although the PM backlog is considerably more significant than the target level, Equinor points out their tendency of generating and scheduling PM activities at the beginning of each year. The clustered generation leads to an instant increase in the backlog and a more extensive backlog than a leveled generation. Equinor also points out difficulties in reducing the backlog due to the amount of work scheduled during production stops. Furthermore, the report also indicated a fire-fighting reality driving the maintenance management, with an unclear assessment of the consequences of such management principle. Difficulties with managing the maintenance according to the maintenance plan due to unexpected events are also pointed out in the recent accident report from Equinor's facility at Hammerfest LNG (Hallan et al., 2021).

1.2 Problem description

The rapid development of technology has a substantial impact in the industry. The items becomes more complex and dependent on mechatronics. The mechatronic units is increasingly connected with information systems, generating an enormous amount of operational data. Furthermore, the operational data is starting to become available to the front line workers at site via hand held devices. This gives them real-time information about the state of the fleet, procedures and maintenance jobs. Furthermore, it enables them to immediately update maintenance records, and create new maintenance suggestions at site. Despite the increasing availability of information and tools, reports shows that managing maintenance is a complex challenge. A logical question is how to utilize technological developments to better manage maintenance. How such tools can improve the capacity, reduce maintenance backlog, and prevent hazard events.

Instead of looking at the symptom and answer the question how to manage maintenance better, we take a step back and start investigating the question why. This project is a preliminary investigation into the question why there is such a challenge with managing maintenance, despite the promising development within technology. We start the investigation by study features of the maintenance process and backlogs. The investigations are carried out via computer modelling and simulation.

The contribution of this project is thus twofold. First, we develop a proof-of-concept maintenance modeling and simulation tool using a combined framework of discrete event simulation and system dynamics. Secondly, we use this tool to investigate features of the maintenance process and backlogs.

1.3 Thesis structure

The remainder of this thesis is structured as follows. The following section presents the relevant theory behind the developed tool. Subsequently, section 3 presents the method behind the development of the tool and the experiments. Section 4 present the experimental protocol, followed by five different experiments performed using the developed tool, including a brief discussion of the results. The discussion of the results continues in section 5 where the results are compared with similar works. Lastly, section 6 summarizes the work and gives some suggestions for interesting future works.

2 Theory

A variety of different terms and concepts are applied in this report. This section provides a more detailed description of some of the important concepts.

2.1 Maintenance

Maintenance is defined according to CEN (2017) as "the combination of all technical, administrative and managerial actions during the life cycle of an item intended to retain it in, or restore it to, a state in which it can perform the required function." There are several types of maintenance defined in CEN (2017). The standard separates maintenance aimed at improving an item by changing the intrinsic dependability characteristics and maintenance aimed at retaining and restoring the condition of an item. The latter is split into preventive maintenance (PM) and corrective maintenance (CM). PM is further split into predetermined and condition-based maintenance. Although preventive maintenance is "maintenance carried out intended to assess and/or to mitigate degradation and reduce the probability of failure of an item" (CEN, 2017), in this report, both preventive and corrective maintenance is assumed to be able to restore an item into perfect condition. The maintenance types implemented in the developed tool are shown in fig. 1. The corrective maintenance is assumed to be unplanned and thus to be more resource-demanding than preventive maintenance.

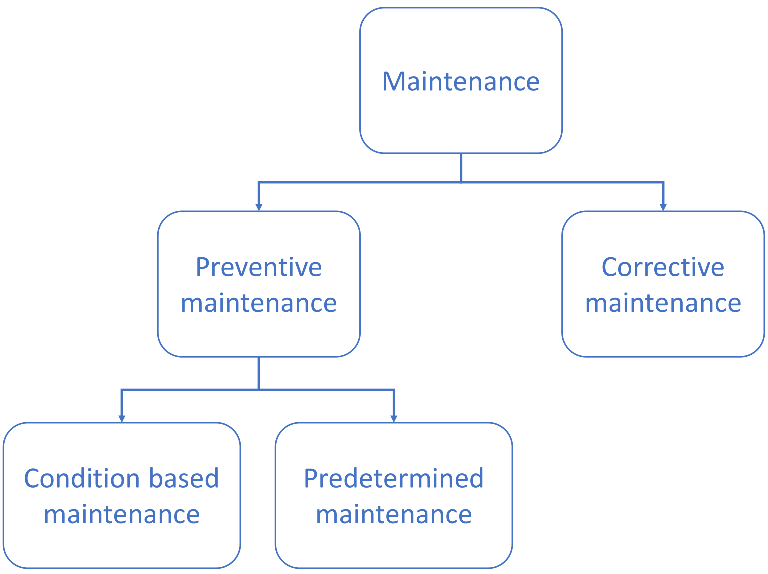

Figure 1: The implemented maintenance types in the modeling and simulation tool. Figure derived from CEN (2017).

While maintenance is a technical problem of retaining and restoring an item, managing maintenance is a socio-technical challenge. The challenge involves a technical part such as determining maintenance programs and strategies and a social part involving operational decisions, bounded rationality behind them, and interactions with parts that affect and are affected by the decision. Maintenance management is defined as "all activities of the management that determine the maintenance requirements, objectives, strategies, and responsibilities, and implementation of them by such means as maintenance planning, maintenance control, and the improvement of maintenance activities and economics" (CEN, 2017).

A maintenance backlog is defined by Rødseth and Schjølberg (2017) as "the amount of unfulfilled demands at a given point of time in explicit reference to predefined standards to be achieved." In this report, we distinguish and have different standards regarding the PM and CM backlog. The PM backlog consists of all the work orders that are due at a particular point in time. The backlog indicates the number of choices a maintenance planner has to evaluate when generating the next maintenance plan. Unfulfilled work orders are rescheduled based on a delay.

On the other hand, the CM backlog consists of all items that are in a failed state. The CM backlog is measured via a notion of risk exposure, which is in this report defined as the number of failed items divided by the total number of items at a given time 1. The measurement assumes that all the items and their failure are equally contributing to the risk.

$$\text{Risk exposure}(t) = \frac{\text{Number of failed items}(t)}{\text{Total number of items}} \tag{1}$$

2.2 System dynamics

System dynamics (SD) is a methodology to build models to enhance learning and the understanding of complex systems (Sterman, 2000). The goal is to reveal the system structure to understand better why specific patterns and events are occurring, as well as being able to "find management policies and organizational structures that leads to greater success" (Forrester, 1961, p. 449). The main concepts within SD models are feedback loops, stocks and flows, and delays.

Feedback loops

One of the first steps to understand complex systems is to discover and represent the interactions among the components of a system. These interactions are in the world of SD called feedback. The feedback structure is viewed as one main contributor to the complex behavior that arises in the system (Sterman, 2000, chap. 4). Feedback is a directed interaction between two components and is either positive or negative. Positive feedback means that an increase of the first component leads to an increase in the second. While for negative feedback, an increase in the first component leads to a decrease in the second. The dynamics of the system arise from feedback loops. These loops are either positive (reinforcing) or negative (balancing). Positive feedback loops tend to reinforce whatever is happenings in the system. An increase in the capacity needed to perform corrective maintenance decreases the total available capacity of the maintenance team, which decreases the capacity available to perform preventive maintenance, which again increases the number of failures, which increases the first variable, and the loop starts over again. However, the reinforcement loops can be either virtuous or vicious (Sterman, 2000, chap. 4). The above-mentioned positive feedback loop is a vicious cycle where the amount of failed items increases. However, the cycle is virtuous if the capacity needed to perform CM declines. In that case, the same feedback loop will further decline the capacity needed to perform CM, under the assumption that PM can prevent failures. Finding policies that turn vicious cycles into virtuous of better performance is one of the goals within SD (Sterman, 2000, chap. 4).

A feedback structure representing a "shifting the burden" archetype in a maintenance system is shown in fig. 2. In this structure, there are two balancing feedback loops, *CM*, *PM*, and one reinforcing feedback loop, *CM eats up PM*. The latter has the unintended effect of shifting a system in an undesired direction. In the experience of a problem's symptom, i.e., a high amount of defect items, there are two ways to cope with the problem. One symptomatic and one fundamental solution (Größler, Thun, and Milling, 2008). The first is a response to the symptom: perform CM to maintain the failed items, while the latter is to perform PM to prevent failures from occurring. While the first has instant feedback in terms of fewer failed items, there is a delay in the latter. In a maintenance system, these two responses are connected due to the limited capacity of the maintenance department. By prioritizing repairs and disregarding preventing failures, at first, the number of failed items declines. However, the capacity to perform PM is reduced as well, resulting in even more failures. This unintended effect of prioritizing the symptomatic solution over the fundamental solution is typical in an overloaded maintenance department (Größler, Thun, and Milling, 2008).

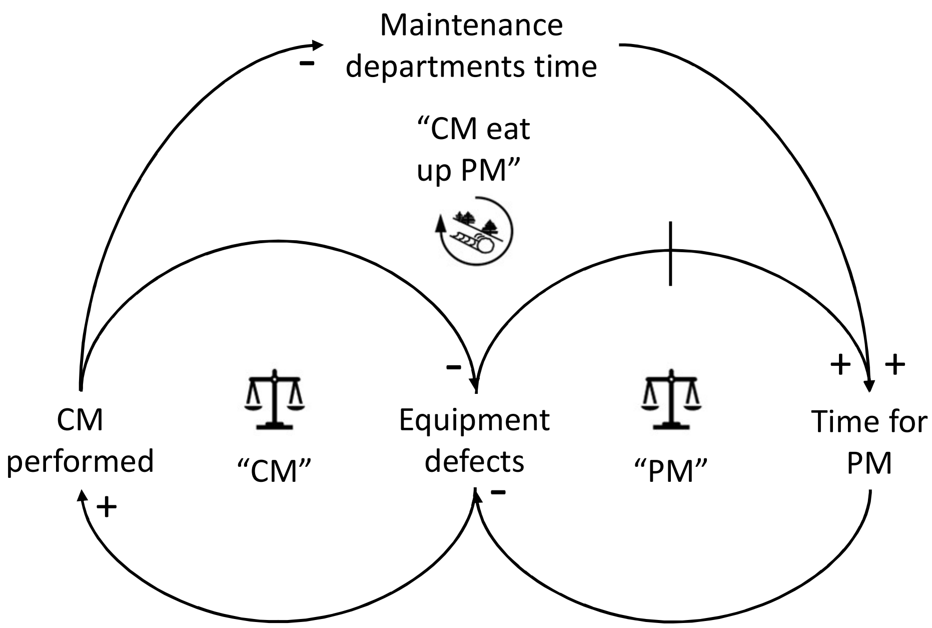

Figure 2: "Shifting the burden" archetype in a maintenance system. Adapted from Größler, Thun, and Milling (2008).

Stocks and flows

Another central idea within SD to understand the dynamics of complex systems is the concept of stocks and flows. Stocks are accumulations and represent the state of the system. The stock accumulates, for instance, material, information, maintenance activities, or competence. The accumulations provide information upon which decisions are based (Sterman, 2000, chap. 6). The stock changes through its connected flows. Flows are generation and completion of activities, acquiring and decay of competence, success, and failures of projects. When the inflow exceeds (is less than) the outflow, the stock will increase (decrease). Although this might seem intuitive, research shows that even highly educated adults tend to misunderstand the behavior of stocks and flows and often misinterpret the behavior of a stock to be positively correlated with its flows (Sterman, 2010).

A stock of maintenance activities accumulates the difference between the rate at which maintenance activities are generated and the capacity to perform maintenance. Having a stock creates a delay between the generation and the completion of maintenance activities. Figure 3 shows a single stock and flow diagram of a maintenance process. There are two ways to decrease a stock. Either by decreasing the inflow rate below the outflow rate or increasing the outflow rate above the inflow rate. Although this is trivial, it is important to recognize the "below" and "above" requirements. A decrease in the inflow does not necessarily imply a decrease in the stock. Furthermore, although this is true for our simple model, considering a systemic perspective, there might be feedback and interaction between the inflow and the outflow. For instance, when completing a maintenance activity, there is a delay before scheduling a new activity for the same item, linking the inflow and the outflow of the backlog.

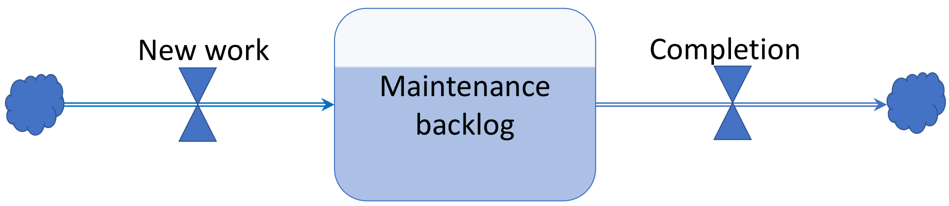

Figure 3: The figure shows a single maintenance backlog stock with in- and outflow of maintenance activities. The stock is the box, and the flows are the pipes. The flow regulators are represented as valves. The "clouds" are the source and sinks for where the maintenance activities come from and disappear.

Delays

As mentioned in section 2.2, stock creates delays in processes. The concept of delays is the third central idea that impacts the behavior of the system the most. Often the cause and effect are thought of as closely linked in time and space. This idea of cause and immediate effect comes from experience and linear thinking. Touching a hot stove gives immediate feedback. There might, however, be a substantial delay between the cause and the effect in complex systems, making it hard to visualize and capture the behavior of the causal relationships. There might be substantial delays when introducing new policies, concepts, and technology into a maintenance process before the effect appears. It takes time to readjust the operation and change the culture and behavior. At first, the effect might appear to be a less productive process, as the time needed to implement the change reduces the available capacity to perform maintenance, leading to increased backlogs. Underestimation of such behavior and the delays involved are common reasons for the failure of improvement initiatives (Valerdi and Fernandes, 2011).

2.3 Models

A model is a simplification of the system it is representing. If a model is as complex as the system it represents, it will not help to understand the complex system any better than the actual system itself (Sterman, 2002). If a map, for instance, were as complex as the world, it would not be any more helpful than the world itself. Along with our experience of nature, people and organizations, our mind develops mental models. These models represent how we picture the world. Although such models may be complex and sophisticated, the capacity of the human mind to formulate and visualize the effect of feedback, nonlinearities, and delays to solve complex problems is limited. The human capability even comes to short in simulating a simple first-order linear feedback loop Sterman (2000) (chap. 1). These mental models are the foundation of pragmatic and formal models. While the first is primarily aimed at facilitating communication, the latter is often used for calculation purposes or computer simulations (Rauzy and Haskins, 2019). To be able to understand the complex interactions and the behavior of systems, Sterman (2000) (p. 37) argues that simulation studies are essential.

2.4 Discrete event simulation

Discrete event simulation (DES) is a modeling and simulation technique based upon Monte Carlo simulation, meaning that the simulation is performed multiple times (Brailsford, Churilov, Dangerfield, et al., 2014). The essence of DES is that events occur in discrete time steps, which creates a jump in the time domain from one event to the next. The firing of an event changes the state of the model, while the state is constant in between the events. A DES model consists of entities, queues, activities, and resources. In a maintenance system, the entities are the items, queues are the maintenance backlogs, activities are the maintenance work, and the resources are the maintenance teams. In a simulation run, entities flow through a network of activities and wait in queues in between. fig. 4 shows a cyclic discrete-event perspective of a maintenance process. The process is, in essence, similar to the stock and flows model in section 2.2. However, in this model, the inflow and outflow are connected. After the execution of maintenance, with some delay, the activity is due again. This may, for instance, be according to an interval or the condition of an item. The maintenance backlog is the queue caused by the accumulation of pending work. As described in section 2.2, such accumulation causes a delay in the process. The cycle time for a maintenance activity is a combination of the traveling time around the circle and the delay caused by the queue. In a maintenance system, the queue may be prioritized according to heuristics, e.g., criticality, and not according to a first-in-first-out principle illustrated in fig. 4. Furthermore, the activities may be of different criticality, duration, travel time, and type—for instance, preventive, corrective, and modification activities.

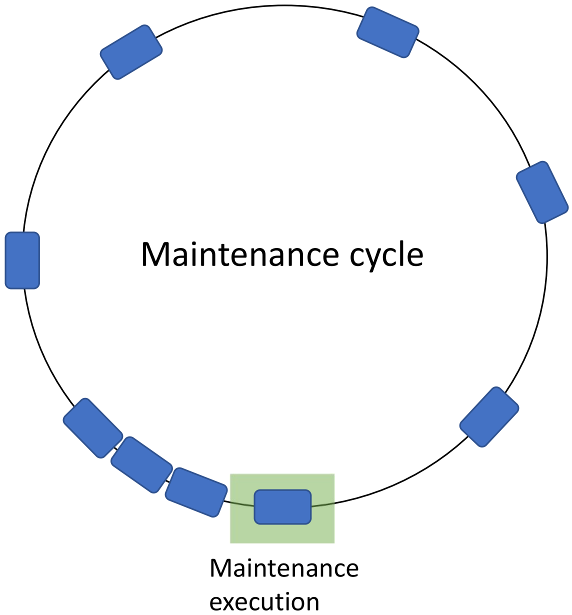

Figure 4: The figure shows a cyclic discrete-event perspective of a maintenance process. Maintenance activities are flowing around the cycle. After some time, the activities are due for maintenance. The capacity to execute maintenance causes a queue of maintenance activities.

3 Method

In this project, we have developed a proof-of-concept modeling and simulation tool. The simulator is developed based on our mental model of the maintenance process and discussions with Equinor, and is intended to study how capacity, backlog, PM program, and accuracy in the planning process are connected and dependent. We apply the developed modeling and simulation tool to perform scientific experiments through computer simulation. We iteratively increase the complexity and adjust the model's parameters. The results of the experiments serve as a base for further investigation in the real system. The investigations aim to find mechanisms in the real system preventing or amplifying the behaviors the model produces. The investigation leads to further improvements of the model, new experiments, and investigations.

The modeling and simulation tool is based on systems dynamics and discrete event simulation principles and is developed using Python programming language. Fundamental in the tool is the creation of models. Although a model by definition is a simplification of the system it represents, we create a metaphor model of the system in our project. The goal of the model is to serve as a learning laboratory used to investigate scenarios, study different decisions and policies, and capture possible behaviors of the real system. To emphasize this, the model is created to operate in units of time rather than, for instance, hours. Furthermore, multiple variables are considered static rather than dynamic due to a lack of information. Instead of modeling the variables dynamically, we investigate how the variable in its extreme affect the system, for instance, the prioritization accuracy of PM activities. Although this variable might be dynamic and dependent on the number of activities to prioritize, we consider it static during a simulation run. The extreme scenarios indicates possible effects of the variable, and thus if investigations into the dynamics of the variable should be explored.

Due to the introduced randomness in policies and information, we perform Monte Carlo simulations. The results are the averages of the simulations, as we are interested in capturing the average behavior of the system and not the behavior due to one specific realization. To reproduce the experiments, a random seed is set before simulating each scenario, which is the only truly random variable of the pseudo-random generator of Python. The seed enables to reproduce the random sequence, and thus the experiments.

The modeling and simulation tool

The modeling and simulation tool supports the creation of a fleet containing an arbitrary number of independent items with one failure mode. The items themselves are unique, with an arbitrary number of possible states. The degradation distribution in between the states is unique for every transition. Implemented in the tool are deterministic transition and transitions based on empirical distributions. This enables the possibility to fine-tune the degradation process of items to hypothesis and historical data. Fundamental in the simulation tool is a DES engine that drives the simulation process. It draws the degradation events for the items in the fleet and drives the maintenance process. The maintenance process consists of generating a maintenance plan for each time unit, based on the policy, CM backlog, scheduled PM activities, and the available capacity. Scheduled PM activities that are not added to the plan are rescheduled based on a given delay. The PM activities can either be generated based on an interval or based on the item's condition and a maintenance threshold. An illustration of the maintenance process implemented is shown in fig. 5. Other influential factors to the maintenance process, such as the failure criticality, tools, work hours, production, spare part policy, workarounds, improvement initiatives, and competence, are considered outside the model's boundary.

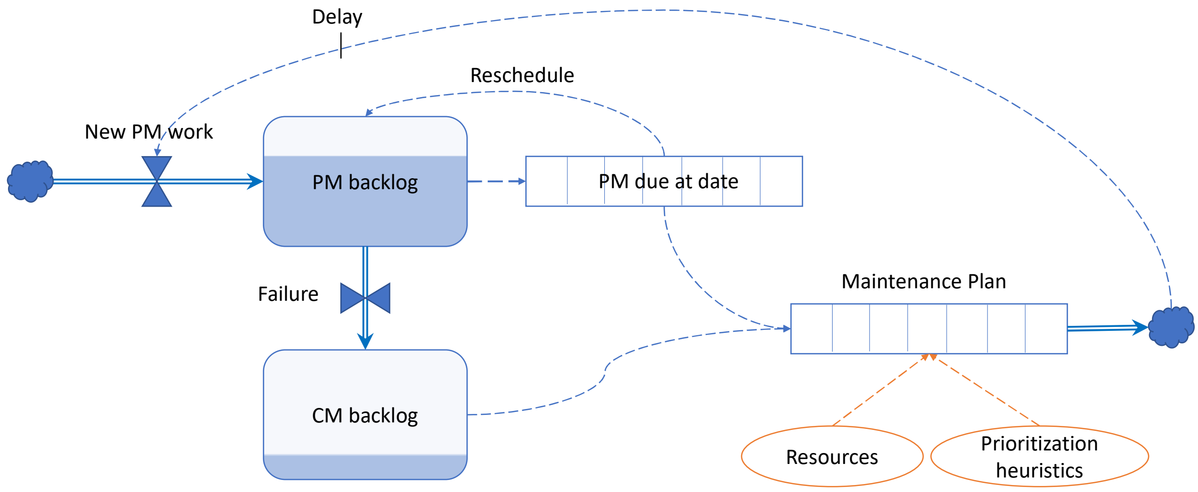

Figure 5: Illustration of the maintenance process implemented in the modeling and simulation tool. The valves in the simulation model illustrate flow of tasks, the clouds illustrate all the items not in the backlogs, while the boxes illustrate the backlogs.

Central in the modeling and simulation tool is the prioritization policy of the PM activities. There is an underlying assumption that the earliest created PM activity is the most important in the prioritization policy implemented. This aligns with a mental model of a deterministic degradation process and an interval-based PM policy. Thus, the item with the earliest created PM activity is the most degraded. Although the assumption is true in the following experiments, it may not be true when considering a stochastic degradation process or considering a mixture of interval and condition-based PM activities. In such scenarios, there is a need for a more informed prioritization policy.

4 Experiments and results

In the following experiments, we study the behavior of our model under given scenarios. We investigate how the PM programs are balanced with the available capacity, the impact of uncertain information under different policies, and how changes to policies change the behavior. The experiments are carried out by using the developed DES+SD tool to create a model and run simulations, as described in section 3.

The basis of the experiments is a fleet of identical and independent items, with only one failure mode. All items can be in four different states, working, D1, D2, and failed, as shown in fig. 6. The transitions between the states are deterministic to reduce the randomness involved and focus on the behavior caused by the system's structure. The state of the item follows the Markov property (Rausand, 2014, p. 121), meaning that the future state of the item is independent of the past, given the present state.

At each time step, a maintenance plan for the next time step is generated. The created plans are not altered. Thus, if a failure occurs, it does not affect the ongoing plan. Both preventive and corrective actions are considered perfect, i.e., the condition for the item is as good as new after a repair. Both the available capacity for each plan and the cost of performing CM and PM are deterministic. The capacity of a maintenance plan is 10 units, of which the cost of CM is 2 units and PM is 1 unit. Thus, at maximum, it is possible to perform 10 PM activities or 5 CM activities, or a combination of those.

In all the experiments, we study three different prioritization policies between CM and PM work, as described in table 1 to study the effect of prioritizing a certain amount of unexpected failures above the maintenance program.

In the developed metaphor model of a maintenance process, the parameters are selected arbitrarily. The combination, however, is not. We have modeled a process in which the capacity to perform PM is sufficient to keep the fleet of items in a healthy state. The PM program interval is selected based on the idea that the more PM, the better, and that a margin between scheduling the PM and failure of the item yields a flexible organization. The planned maintenance is assumed to be less costly in terms of needed capacity than unplanned. Upon unplanned events, several factors may delay the maintenance process and add capacity cost to the execution. Such as, for instance, the availability of spare parts, competence, tools, safety analysis, and job descriptions. To illustrate this, the capacity cost of CM is twice the cost of PM.

Table 1: Prioritization policies between CM and PM work.

Parameter	**Input**
CM0	PM activities are prioritized above CM
CM50	50-50 CM and PM activities
CM100	CM activities are prioritized above PM

Table 2: Input variables constant for each experiment.

Parameter	**Input**
Maintenance plan capacity	10 units
Cost of PM	1 units
Cost of CM	2 units
$Working \rightarrow D1$	5 time units
$D1 \rightarrow D2$	35 time units
$D2 \rightarrow F$	10 time units
Random seed	1 223 344
Number of trials	1000

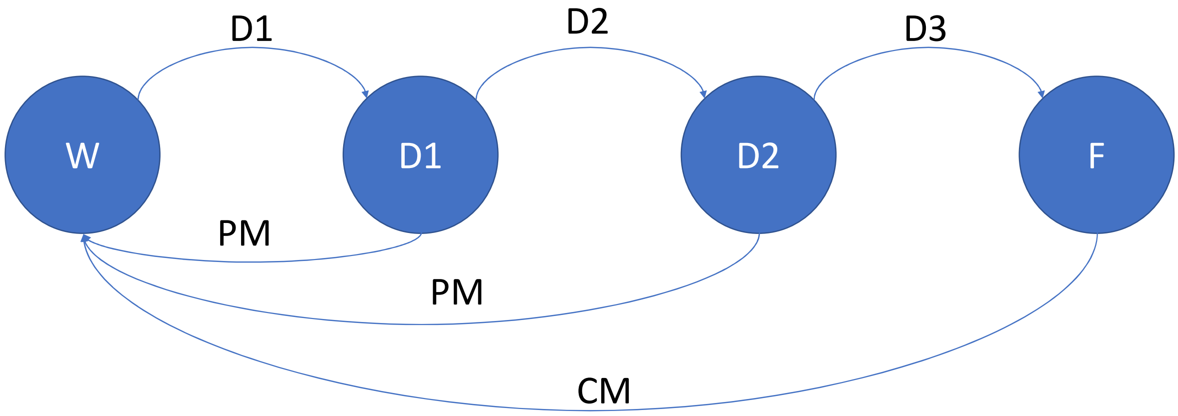

Figure 6: Markov diagram of one item, showing the transitions between the states W, D1, D2 and F. As well as the maintenance type applied to repair the item.

4.1 Experiment 1 - Base case

In this experiment, serving as the base case for further experiments, we investigate the hypothesis that "interval-based PM programs will adjust themselves to the capacity available." The experiment is based on a fleet of 500 items, of which all are initially working. The items are added into the system with a pace of 10 items per time unit, equal to the capacity to perform preventive maintenance. Furthermore, after a delay of 30-time units, PM is scheduled according to the PM interval. Items are thus introduced into the system from time step 1 until time step 50, while maintenance for those items will be scheduled from time step 30 until time step 80. Preventive maintenance is assumed to restore the item into perfect condition. The preventive maintenance is performed sequentially with a first-in-first-out principle, which implies that the most degraded item is maintained first due to our deterministic degradation. In this experiment, there is no randomness involved, and hence no need to perform the simulation multiple times to average the different realizations of random processes.

Table 3: Input parameters to the base case experiment.

Parameter	**Input**
Number of items	500
Rescheduling delay	1 time unit
Accuracy in PM prioritization	100%
PM interval	30 time units
Number of simulation runs	1
Mission time	200

Results

The simulation results are shown in fig. 7, fig. 8 and fig. 9. In this experiment, the preventive maintenance program can keep the entire fleet in a healthy state. As seen in fig. 7, the stock of due PM activities increases from 0 to 10 at time 30. In this period, the flow into the stock is the scheduled activities from the program's initialization, while the outflow is the preventive maintenance activities performed. Since the inflow is equal to the outflow, the level of the stock is stable. However, in the period 60 to 80, the inflow consists of two sources. Both scheduled activities from the program's initialization and the maintenance program of those items already have maintained flow into the backlog. Thus, we can perform 10 preventive maintenance actions for every time step in this period, while 20 new preventive maintenance activities are due. The backlog increases due to the difference between the inflow and outflow. At time 80, however, the inflow equals the outflow, and the backlog is stationary. The average PM interval is shown in fig. 8. The graph displays a similar behavior compared with fig. 7, which is expected as the backlog acts as a delay between the generation and execution of tasks. In the period 30 to 60 the number of PMs due is equal to the capacity, and correspondingly the actual PM interval is equal to 30, the interval in the PM program. The PM interval increases while the backlog increases. However, it increases until time 100 and stabilizes with 50 as the actual PM interval. The actual PM interval corresponds with a calculated time needed to maintain the fleet using preventive maintenance. There are 500 items, and the capacity to perform PM is 10 items per time unit. Thus, the time needed to maintain the fleet is 50-time units, equal to the measured actual PM interval.

The imbalance between the capacity of the maintenance team and the PM program comes at the cost of rescheduling. In this experiment, the expired PM activities are rescheduled to the next time step. As the backlog of due PM items increases, the delay caused by the backlog increases and thus also the average number of times the activities performed are rescheduled, as seen in fig. 9.

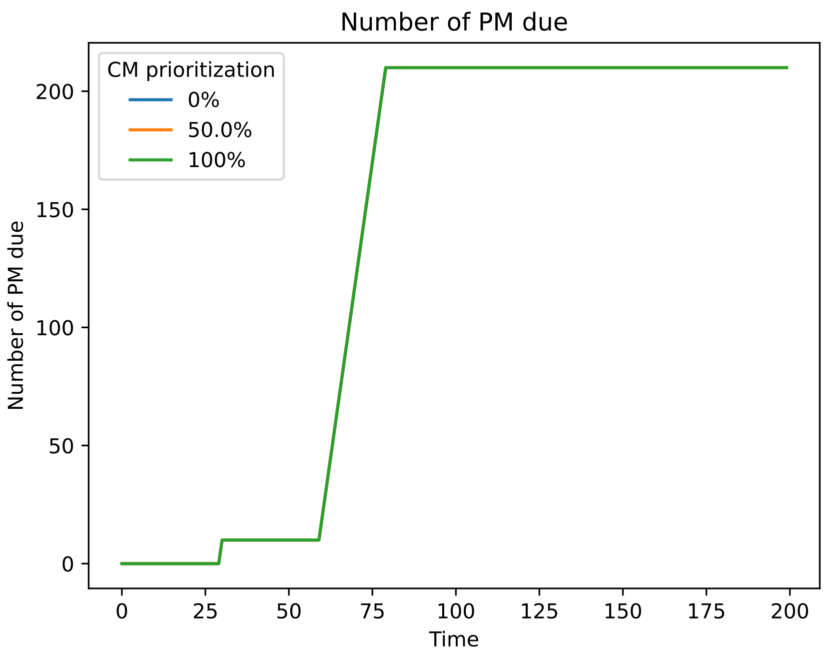

Figure 7: Evolution of the number of due PM at each time unit.

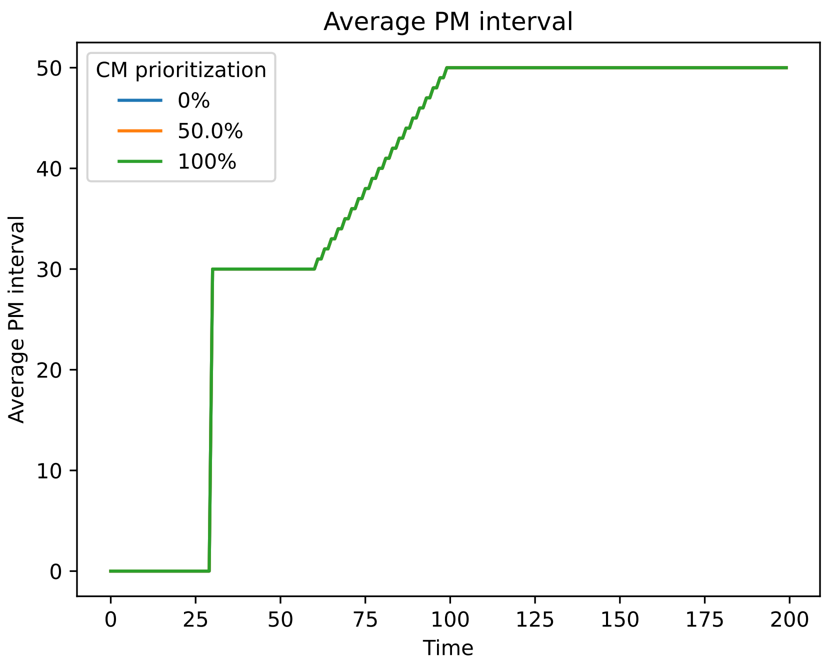

Figure 8: Average of the actual PM interval of the activities performed at each time unit.

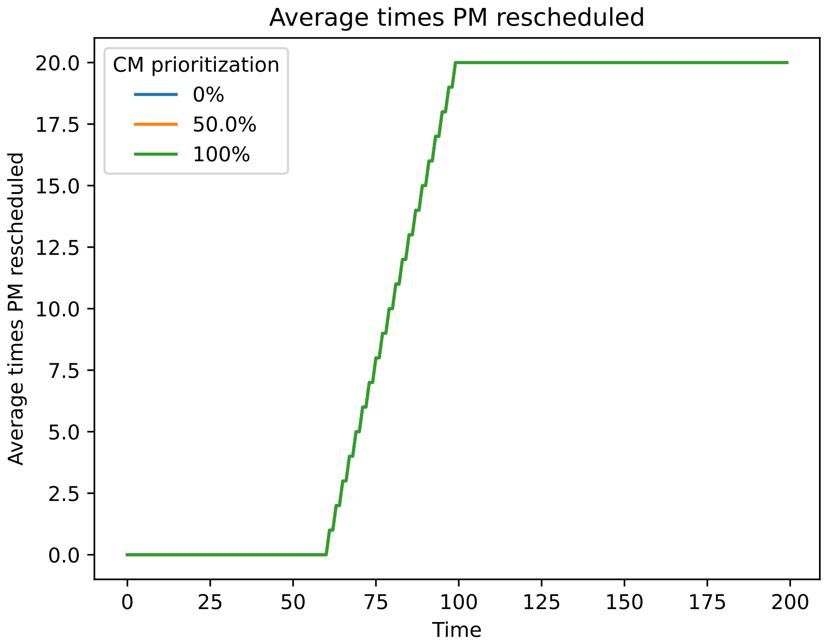

Figure 9: Average of the number of times the PM activities performed are rescheduled at each time unit.

Discussion

In this experiment, the degradation is deterministic, and the preventive maintenance is performed sequentially. Due to the imbalance between capacity and the PM program, a backlog of PM activities arises. The actual PM interval increases to match the capacity to perform the maintenance. The number of rescheduling times is the difference between the actual PM interval and the interval in the maintenance program due to the rescheduling policy. The backlog increases more rapidly than the measurement of the actual PM interval. The actual PM interval measures the difference between execution and creation of the activities executed at each point in time. Changes in the backlog do not appear in the interval measurement until execution of those activities, causing the change when performing maintenance sequentially.

This experiment indicates that there is a balance between PM programs and the capacity available when having limited capacity. This comes at the cost of backlog and rescheduling, where the backlog acts like a delay which increases the actual PM interval.

In this experiment, we have not included the cost and capacity to perform rescheduling and possible risk analysis and the quality of such against the amount of work. Furthermore, we assume maintenance is always possible to perform whenever the maintenance planner chooses to, independent of the availability of competence, the item itself, tools, spare parts, etc. Including such factors creates additional delays in the process. In this experiment, we consider a highly optimized maintenance process, where the capacity to perform maintenance is equal to the actual demand. Due to the deterministic degradation, we are able to keep the fleet in a healthy state, performing maintenance sequentially.

4.2 Experiment 2 - Increase PM interval

In this experiment, we investigate the hypothesis *increasing the PM interval decreases the PM backlog*. The experiment is similar to the base case. However, at time 150, the PM interval for all the items is changed from 30 to 35-time units. In this experiment, there is no randomness involved, and hence no need to perform the simulation multiple times to average realizations of random processes. The input parameters are shown in table 4.

Table 4: Input parameters to the unexpected failure experiment.

Parameter	**Input**
Number of items	500
Rescheduling delay	1 time unit
Accuracy in PM prioritization	100%
PM interval ($0 \rightarrow 149$)	30 time units
PM interval ($150 \rightarrow 300$)	35 time units
Number of simulation runs	1
Mission time	300

Results

The simulation results are shown in fig. 10, fig. 11 and fig. 12. The experiment, and thus also the results, is equal to the base case experiment in section 4.1 until the change of PM interval at time 150. In the period 180 to 184, the number of due PM activities declines from 210 to 160, as shown in fig. 10. At the end of this period, fig. 12 shows a rapid decline from 20 down to 15 in the average number of times PM are rescheduled. There is, on the other hand, no difference between this and the base case experiment when it comes to the average actual PM interval, as shown in fig. 11. Neither is there any difference in the number of PM activities performed in a simulation run, both equal to 2700.

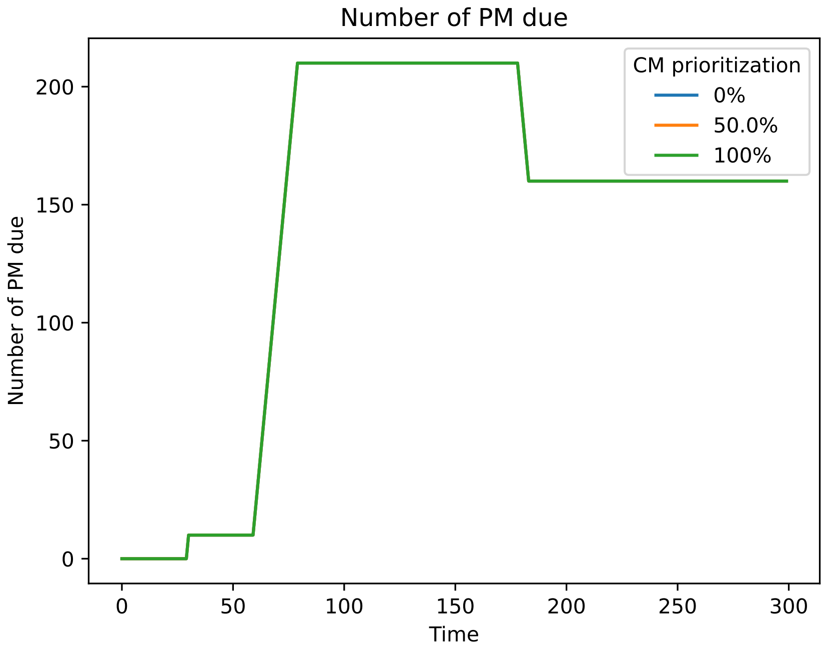

Figure 10: Evolution of the number of due PM at each time unit.

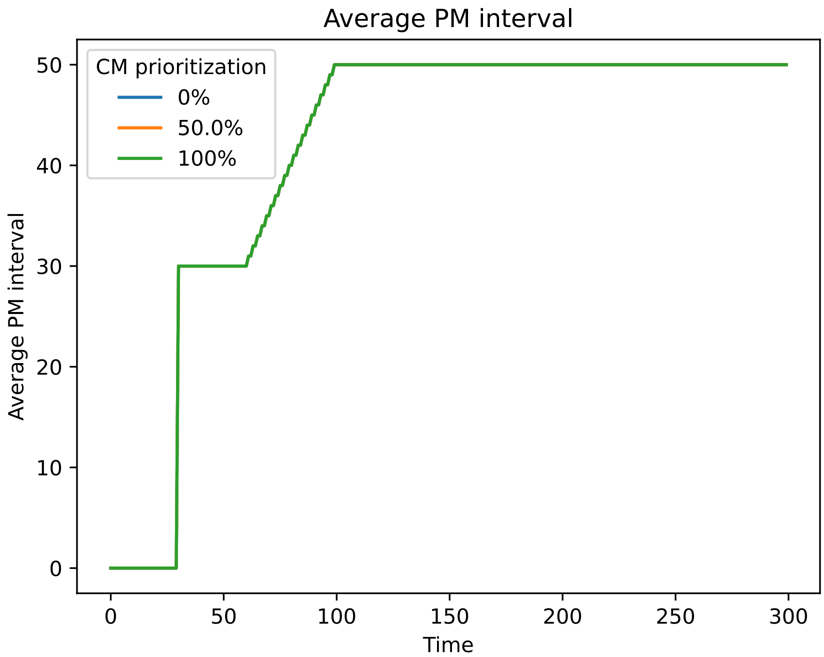

Figure 11: Average of the actual PM interval of the activities performed at each time unit.

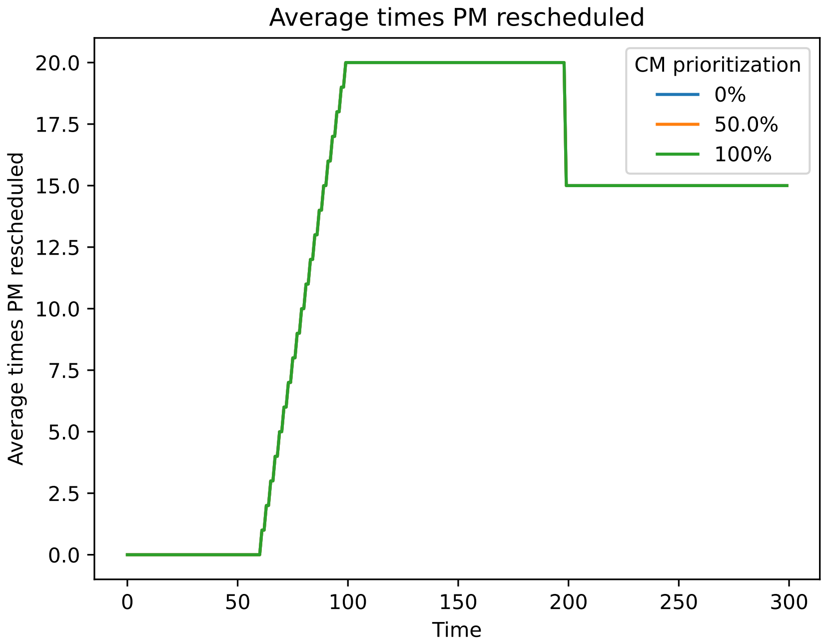

Figure 12: Average of the number of times the PM activities performed are rescheduled at each time unit.

Discussion

In this experiment, we have shown that increasing the PM interval reduces the backlog and the number of times PM is rescheduled. Thus, increasing the PM interval reduces the time a PM activity is in the backlog. However, the increase does not affect the actual PM interval and the number of PM activities performed. The increase of 5-time units in the PM interval reduces the time in the backlog by 5-time units. There is a delay between the change in policy and the measurement of the effect. In our deterministic model, the maintenance is viewed as a cyclic process, as shown in fig. 4. Changing the PM interval causes a gap in the inflow to the backlog. This gap starts when the activities should have been scheduled following the previous interval and last until they are scheduled. Thus, there is a 5-time unit gap where the inflow of the backlog is zero, while the outflow remains the same. During the gap, the outflow reduces the backlog by 50 activities. The reduction in the number of times maintenance activities are rescheduled is not shown until the execution of the first activities following the new policy. The delay is due to the measurement method, where the measured average is the activities fired at a given point in time.

Even though the performance was not affected by the increase of the PM interval, the change might have other effects not included in the model. There might, for instance, be a relationship between the amount of PM due and the accuracy in the prioritization. Further on, there might be a limited capacity to perform rescheduling and risk analysis. A reduction in the amount of rescheduling work reduces the workload, which might improve the quality of the remaining work. The experiment also shows a delay between changing the PM interval and the effect of the change.

4.3 Experiment 3 - Increase Capacity

In this experiment, we investigate the hypothesis *increasing the capacity decreases the PM backlog*. The experiment is similar to the base case. However, at time 150, the maintenance plan capacity is changed from 10 to 11 units. In this experiment, there is no randomness involved, and hence no need to perform the simulation multiple times to average the realizations of random processes. The input parameters are shown in table 5.

Table 5: Input parameters to the unexpected failure experiment.

Parameter	**Input**
Number of items	500
Rescheduling delay	1 time unit
Accuracy in PM prioritization	100%
Maintenance plan capacity ($0 \rightarrow 149$)	10 units
Maintenance plan capacity ($150 \rightarrow 300$)	11 units
Number of simulation runs	1
Mission time	300

Results

The simulation results are shown in fig. 13, fig. 14 and fig. 15. The experiment, and thus also the results, is equal to the base case experiment in section 4.1 until the change of capacity at time 150. In the period 150 to 180, the number of PM activities due declines from 210 to 175, as shown in fig. 13. The average actual PM interval declines from 50 to 45 in the period 150 to 195, as shown in fig. 14. Figure 15 shows that in the same period, the average number of times PM is rescheduled decreases from 20 to 15. The number of PM activities performed in a simulation run is 2850.

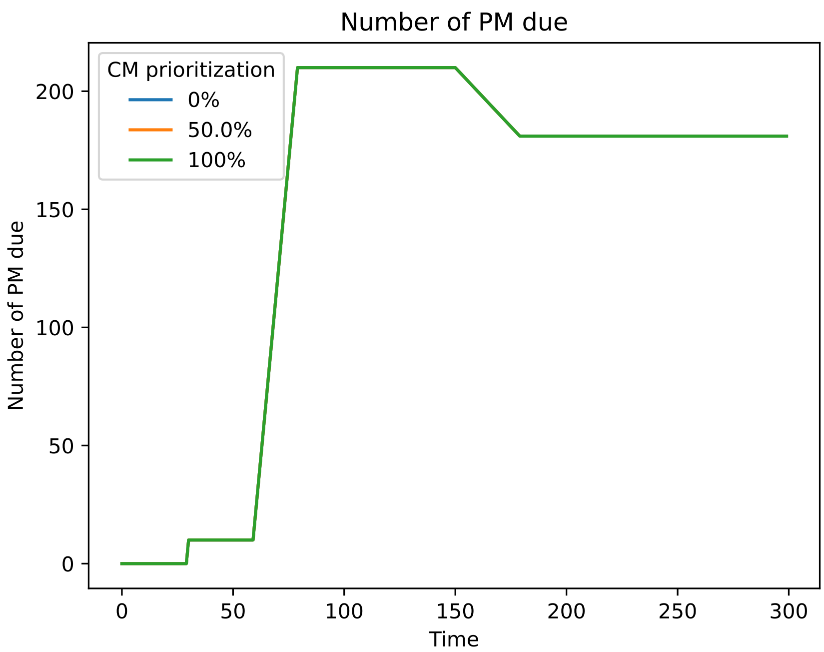

Figure 13: Evolution of the number of due PM at each time unit.

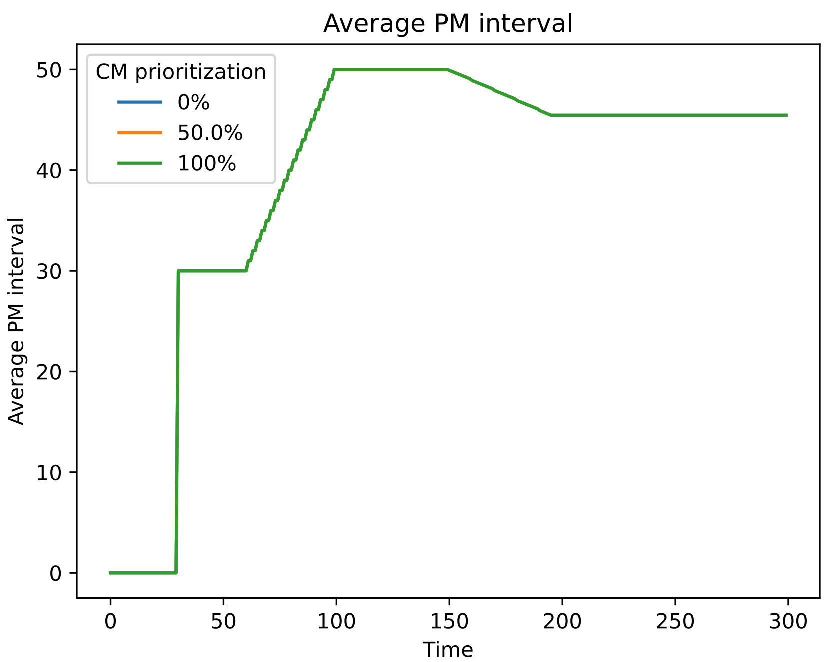

Figure 14: Average of the actual PM interval of the activities performed at each time unit.

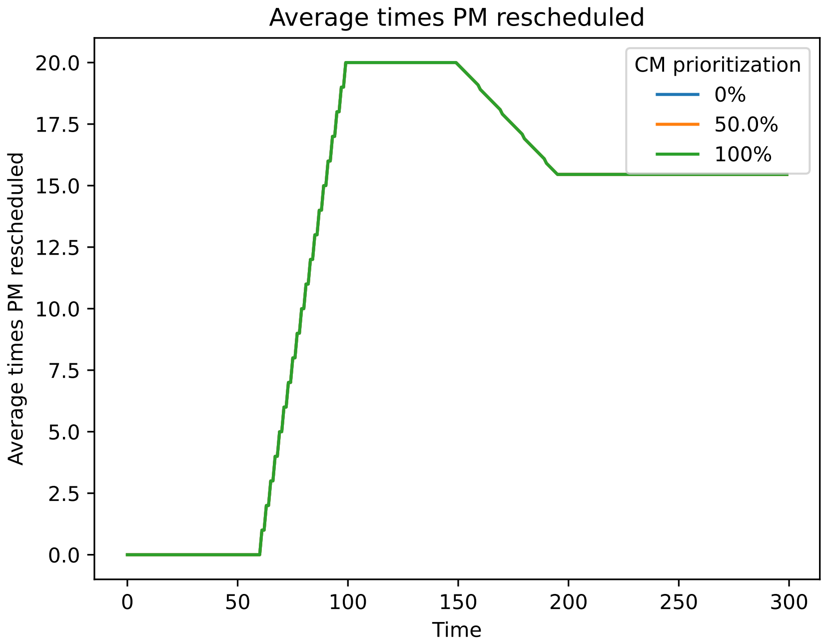

Figure 15: Average of the number of times the PM activities performed are rescheduled at each time unit.

Discussion

In this experiment, we have shown that increasing the maintenance capacity reduces the backlog, the number of times PM is rescheduled, and the actual PM interval. On the other hand, the increase of capacity also increases the number of PM activities performed, from 2700 in the base case experiment to 2850. The increase is due to the reduction of the actual PM interval, which in the cyclic perspective shown in fig. 4 reduces the cycle time, resulting in more activities executed. The increase in maintenance performed may contradict other organizational objectives, such as reducing maintenance costs.

Contradictory to the previous experiment, there is an immediate response to the increase of capacity. However, there is a delay before the full effect of the change is measured. The number of PMs due is viewed as a stock. At time 150, the outflow rate is changed to 11, while the inflow rate remains at 10. The increased outflow reduces the stock. With a delay of 30-time units, equal to the PM interval, the inflow rate is increased to 11, which stabilizes the stock at a new equilibrium. The period of decline in the actual PM interval and rescheduling is even longer. It requires a turnover of the backlog to measure the full effect of the capacity change. When the change occurs, the activities in the backlog have already been rescheduled and delayed using the previous policy. Therefore, the full effect is not shown until the execution of the first maintained activities with the new capacity.

In the experiment, we assume an immediate increase of capacity. The capacity increase may be due to hiring, grouping maintenance, or technological developments. The assumption of immediate implementation is unlikely. New hires require training, and changing the way of working might require internal training. At first, such changes might reduce the capacity to perform maintenance before the intended increase, as a part of the capacity is initially spent on training instead of execution. Adding the before-mentioned to the model would cause a worse-before-better behavior. Underestimation of such behavior is a common reason for the failure of an improvement initiative (Repenning and Sterman, 2003).

4.4 Experiment 4 - Unexpected failure

In this experiment, we investigate how an optimized maintenance process responds to an unexpected failure. The process is optimized in regards to capacity. The capacity to perform maintenance is equal to the capacity needed to perform maintenance just in time before failure. The experiment is similar to the base case. However, the last item in the sequence is initially in state D1. Due to the duration of the maintenance sequence and a maximized actual preventive maintenance interval, this item fails before the PM action takes place. We investigate the behavior of the system with three different prioritization schemes for the distribution of PM and CM activities at the maintenance plan, CM0, CM50, and CM100, as described in **??**. The input parameters are shown in table 6.

Table 6: Input parameters to the unexpected failure experiment.

Parameter	**Input**
Number of items	500
Rescheduling delay	1 time unit
Accuracy in PM prioritization	100%
Number of simulation runs	1000

Results

The simulation results are shown in fig. 16, fig. 17, fig. 18 and fig. 19. Initially, the system behaves as in the base case experiment in section 4.1. However, at time 95, there is a slight change in the risk exposure and the number of PMs due, which is shown in fig. 17 and fig. 18 respectively. The results show a huge difference in how the system behaves under the different CM/PM prioritization policies in response to unexpected failure. While the risk exposure slightly increases under the CM0 policy, it drastically increases under the CM50 and CM100 policy, as seen in fig. 17. There is a 5-time unit delay between the failure and the drastic increase in risk exposure. In this period, under the CM50 and CM100 policy, the risk exposure is reduced to 0. The policies manage to solve the immediate failure before the situation worsens at time 100. Although the increased amount of failures is immediately detectable in the measurement of risk exposure, the information is, however, delayed in the measurement of the number of PMs due, as seen in fig. 16. While the rapid growth in risk exposure starts at time 100, the PM backlog starts declining at 130. That is the point in time where the PM activities for the failed items would have been scheduled. However, the failure of these reduces the inflow into the PM backlog to the capacity to perform CM. Although failures occur from time 95, the average PM interval is not affected. In the CM100 policy, at time 100, the average PM interval drops to 0, indicating that non PM activities are performed. For CM0, and CM50 however, the average PM interval remains close to 50. Thus, this measurement

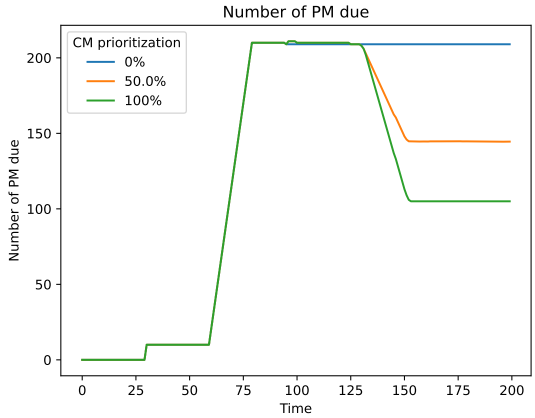

Figure 16: Evolution of the number of due PM.

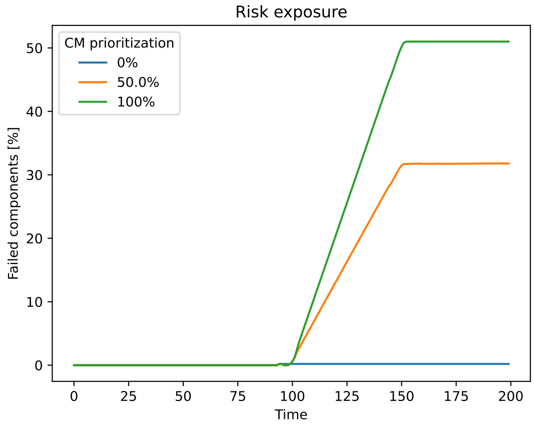

Figure 17: Evolution of the risk exposure.

only indicates the average PM interval for the activities performed. However, looking at the average number of times PM activities are rescheduled in fig. 19, we can see that under CM0, activities are rescheduled 20 times, CM50 20.5 times, and CM100 21 times. This shows that although the maintenance organization performs fewer PM activities under the CM50 policy and none under the CM100 policy, rescheduling the PM activities is still carried out until the items fail.

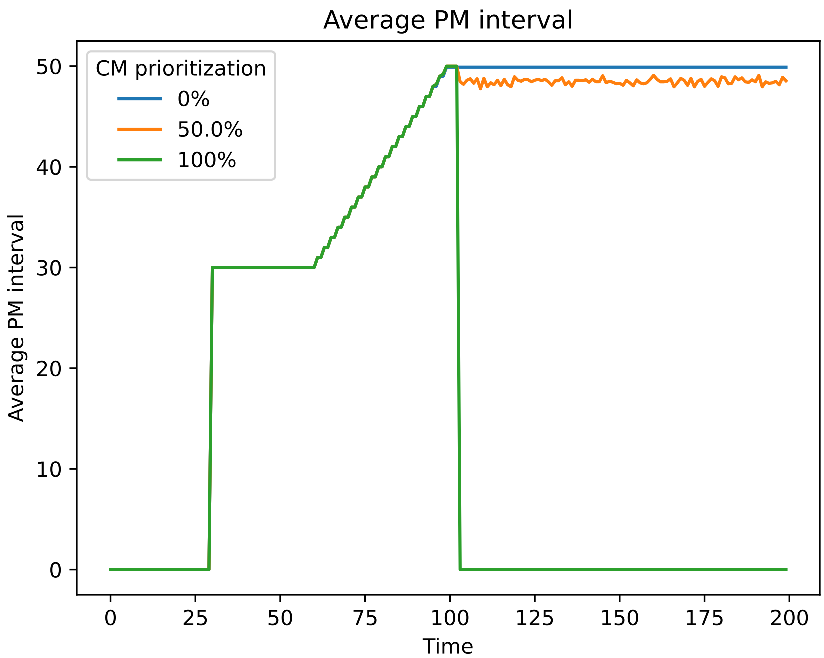

Figure 18: Evolution of the number of due PM.

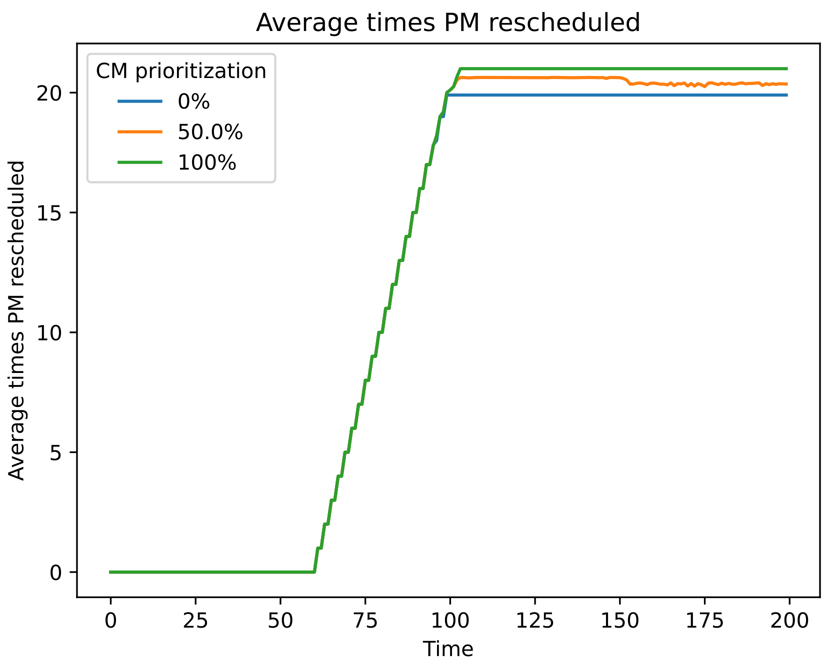

Figure 19: Evolution of the number of due PM.

Discussion

In this experiment, there are multiple factors involved in disrupting the maintenance sequence. First of all, the prioritization of PM activities assumes that the first created activity, and thus the most delayed, are the most critical to perform. This, however, is not the case due to the unit initially in state D1. Furthermore, the just-in-time maintenance leaves no margins for the activities and results in a highly fragile system. Only a slight change disrupts the maintenance plan and causes rapid growth of the risk exposure when prioritizing to maintain failed units. A parallel to the highly optimized just-in-time maintenance is the attempt to implement predictive maintenance. The goal is to predict the remaining useful lifetime and reduce maintenance by performing it just in time before the occurrence of failures. Thus, as our experiments indicate, the implementation of predictive maintenance should evaluate the flexibility of the maintenance process to avoid drastic scenarios due to delays and resource shortages. On the other hand, such drastic scenarios are unlikely to happen. In his research, Morrison (2015) points out that maintenance workers are likely to find workarounds to increase the capacity when facing resource shortages.

The flexibility of the maintenance process could be increased by increasing the outflow capacity. This could, for instance, be achieved by hiring more workers, using overtime, introducing workarounds, technological developments, or group maintenance activities. The increase of the outflow capacity will reduce the delay caused by the backlog, which reduces the actual PM interval and thus increases the margin. On the other hand, a reduction of the actual PM interval increases the amount of PM performed. Thus, an increase in the outflow capacity will increase the amount of PM performed, increasing the cost of maintenance.

Table 7: Input parameters to the base case experiment.

Parameter	Input
Number of items	450
Rescheduling delay	1 time unit
Accuracy in PM prioritization	0%
Number of simulation runs	1000

4.5 Experiment 5 - Random selection of due PM

In the previous experiments, the preventive maintenance activities are performed in perfect order. In this experiment, however, we study the impact of selecting the scheduled PM activities at random. Unlike the previous experiments, there is a margin on the PM activities. The workload is reduced by 10% by reducing the number of items correspondingly. The input parameters for the experiment are shown in table 7.

Results

The simulation results are shown in fig. 20, fig. 21, fig. 22 and fig. 23. Initially, the system behaves as in the two previous experiments. In the period 30 to 60, the rate at which new PM activities are scheduled is equal to the capacity to perform PM, resulting in a stable system with an average PM interval of 30. In the time period 60 to 75, however, the inflow consists of the scheduled activates from the initialization of the program and the maintenance program of those items that we already have maintained. This leads to a rapid increase in the number of PMs due, the average actual PM interval, and the number of times PM is rescheduled, as shown in fig. 20, fig. 22 and fig. 23. At time 80, the risk exposure measurement shows that failures occur, and the risk exposure increases rapidly in both the CM0, CM50, and CM100 policy fig. 21. At approximately time 140, the system stabilized for the CM50 and CM100 policy at a risk exposure of respectively 33% and 48%. For the CM0 policy, on the other hand, the risk exposure still increases, and the number of PMs due and the average PM interval declines at the end of the simulation at time 300. Although the risk exposure increases rapidly in all three policies, the behavior of the PM due backlog is different, as shown in fig. 20.

For the CM0 policy, the PMs due declines at approximately time 80, indicating that the outflow rate is higher than the inflow rate into the backlog. For the CM50 and CM100 policy, on the other hand, the number of PM due increases. This indicates that the flow in is higher than the flow out of the backlog. For the CM50 policy, however, the backlog first increases, then declines before it increases again. At approximately time 125, the number of PM due under CM50 and CM100 plunges until it stabilizes around 110. The average number of times PM activities are rescheduled increases rapidly in all three scenarios. Under the CM100 policy, it increases up to 21, indicating that the PM activities are rescheduled until the items fail. Under the CM0 and CM50 policy, the increase rate starts to decline at approximately time 80. At time 90, the number of times rescheduled decreases. While the number of times rescheduled continues to decrease under the CM0 policy, it increases before it decreases and stabilizes at approximately 15 times under the CM50 policy. The average PM interval shown in **??** increases in the initial 30 at time 60, where the number of PM due starts to increase. The interval increases until approximately time 90, where it drops to zero in the CM100 policy, indicating that zero PM activities are performed. The interval stabilizes at approximately 37 under the CM50 policy, and ending at around 34 under the CM0 policy, still decreasing slightly.

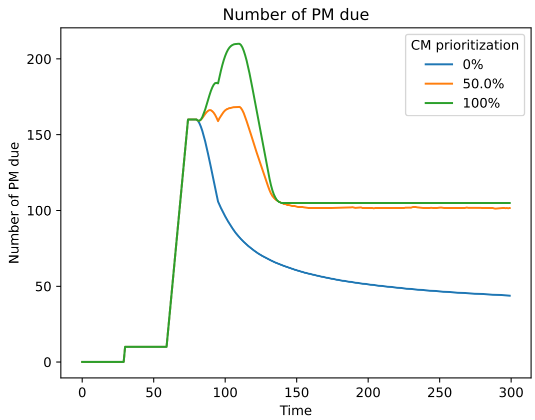

Figure 20: Evolution of the number of due PM.

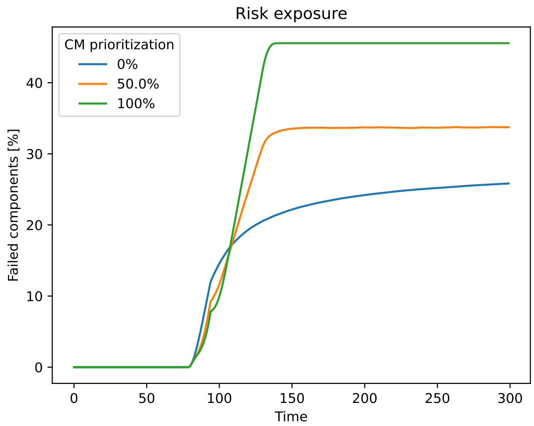

Figure 21: Evolution of the risk exposure.

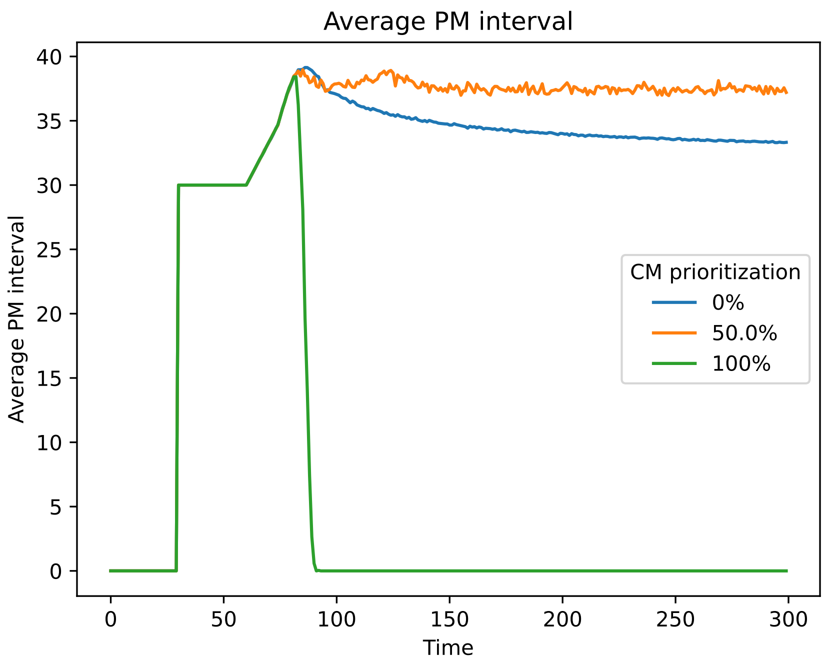

Figure 22: Evolution of the number of due PM.

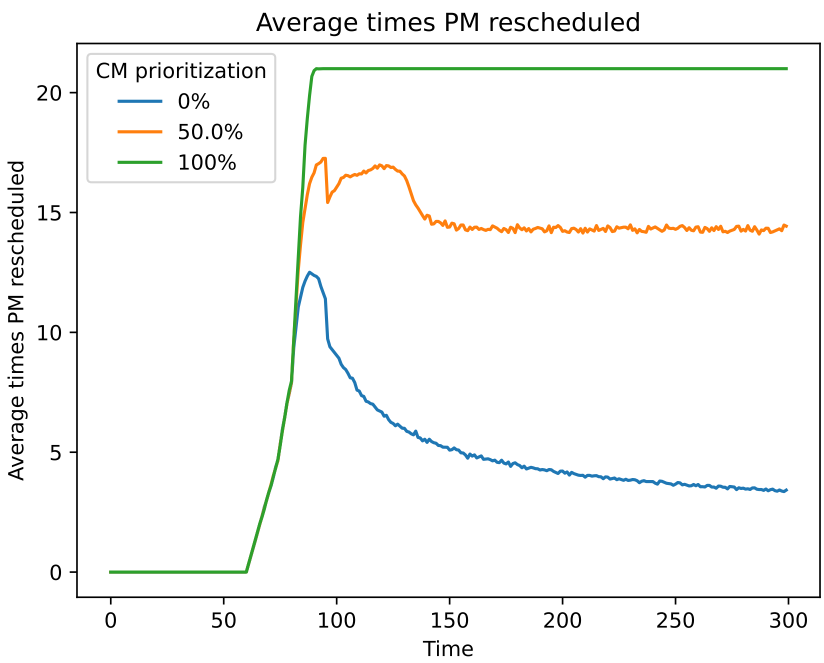

Figure 23: Evolution of the number of due PM.

Discussion

In this experiment, multiple sources of inflow and outflow affect the observed behavior of the PM backlog. Under the CM0 policy, the PM backlog starts decreasing at time 90. The inflow, in this case, consists of PM activities of the items already maintained once. While the outflow consists of the PM activities performed and the failure of items due to the inaccurate prioritization. The difference between the flows results in the observed decline in the backlog. Under the CM50 and CM100 policy, however, the backlog increases. This is due to the reduction of the outflow by prioritizing CM while the inflow remains the same. The number of PMs due stabilizes at approximately 110 under both the CM50 and the CM100 policy. However, the time a PM activity is in the backlog is different, as shown by the number of times a PM activity is rescheduled.

Studying the different in- and outflows of a backlog is essential to understand the underlying behavior of the maintenance system. Our experiment shows that a decreasing backlog might not be due to a more efficient process. Even a stabilized backlog might not indicate that even a single PM activity is performed. These inflows and outflows of the backlog are dependent on the data management and delays in the process. For instance, if PM activities are not removed or rescheduled upon failures, these might contribute to the accumulation of a high PM backlog. Further on, without proper data management, these might also result in performing PM shortly after an item is repaired.

In this experiment, we consider the prioritization accuracy to be constant. It would be interesting to study a potential relationship between the prioritization accuracy and the amount of work in the backlog. Furthermore, if such a relationship between accuracy and backlog exists, it would be interesting to include a PM scheduling policy that tends to cluster the due dates, as mentioned in section 1.1. Having tendencies to schedule the due date of maintenance to a specific point in time, e.g., first in a month, might cause the backlog to have a "saw-tooth" behavior. Thus, cause the backlog oscillate, and periodically be higher than a leveled scheduling policy, and lower otherwise. Additionally, study the impact of data management policies. For instance, study a scenario where PM activities are not removed upon failure, which increases the PM due backlog, and complicates the prioritization work, and might result in PM activities performed shortly after CM for an item.

5 Discussion

Our experiments indicate that one contributing factor to a maintenance backlog is an imbalance between the maintenance programs and the capacity available. The backlog itself causes a delay between the generation and execution of activities, which balances the maintenance programs with the capacity available. We have investigated two ways of reducing the backlog. First, to increase the PM interval, and second, to increase the capacity to perform maintenance. Both approaches reduced the PM backlog. The latter, however, also reduces the actual PM interval, which increases the number of PM activities performed. This might contradict other organizational goals, such as reducing maintenance costs.

A third way of reducing the backlog, as pointed out by Größler, Thun, and Milling (2008), is to reduce the maintenance needed via modifications and improvements. It is essential to reinvest the productivity generated by improvements into further improvements. Attempts to minimize costs by harvesting the improvements via headcount reduction stops a potential reinforcing loop of improvements (Sterman, 2000, chap. 2). In our model, modification and improvement maintenance are not included. However, our experiments show how an increase in the capacity reduces the backlog and the actual PM interval, resulting in more PM performed. Having a prioritization scheme where CM is prioritized followed by PM and, at last, improvements might yield the same result. Thus, any improvement in the performance and capacity is harvested by improvements in CM and PM performance, resulting in no further improvements of the performance.

We have shown that a highly optimized maintenance process, where maintenance is performed just in time, might be highly fragile. A slight increase in the demand might disrupt the maintenance plan and change the system's behavior from a preventive maintenance program to a corrective reality. Lastly, we have shown that inaccurate prioritization of due PM might cause failures and lead to a corrective reality.

The rapid change in the level of risk exposure is unlikely to happen in reality. Mechanisms within the system will most likely prevent such behavior. With the use of system dynamics models, Morrison (2015) studies how a maintenance organization responded to resource shortage with the use of workarounds. He shows via simulations how well-intended workarounds work as both a solution and mask the underlying problems. Further on, while being well-intended to get the job done, the workarounds might erode the process skills or other forms of organizational capabilities, which over time might push the system towards poor performance.

Our modeling and simulation tool is based upon the principles of discrete event simulation. Alrabghi and Tiwari (2015) criticize maintenance optimization studies that use discrete event simulation for considering unrealistic and oversimplified systems. They argue that changes to more realistic assumptions in such models would lead to drastic changes in the outcome, leaving them highly sensitive to deviations from their optimized system likely to occur in the real world. Our modeling and simulation tool share many of the simplifications applied in the optimization studies. However, our tool is not aimed at optimizing maintenance and predict future behavior but rather to serve as a learning laboratory, study possible impacts of decisions and policy changes, and play "what-if" scenarios.

In response to the oversimplified maintenance optimization simulation, Linnéusson (2018) developed a multi-objective maintenance optimization concept using the combined framework of discrete event simulation and systems dynamics. The developed concept enables a holistic view of the maintenance process while trying to optimize the maintenance. This relates to our work by combining the same concepts. However, the work aims at optimizing maintenance rather than investigate maintenance problems.

Carroll, Sterman, and Marcus (1998) have successfully applied system dynamics to create a maintenance game. This facilitates a holistic learning process among different parts of a manufacturing process and how the actions and decisions affect other parts, both in a short- and long-term perspective. The game's goal is to facilitate collaboration between the different parts and work towards a common objective. This game has, among others, successfully been applied to improve the maintenance process at Du Pont, where they used this to change the way maintenance is understood from the top to the bottom of the organization (Carroll, Sterman, and Marcus, 1998). Although our project aims at the same objective as the maintenance game; to create a learning laboratory and facilitate communication among stakeholders, we have developed a proof-of-concept modeling and simulation tool to investigate the maintenance process using computer simulation.

6 Conclusion and further works

This project serves as preliminary work to investigate how SD and DES could be combined to understand industrial maintenance problems better and make risk-informed decisions. We have developed a proof-of-concept modeling and simulation tool using a combined framework of SD and DES. The tool has been used to model a simple maintenance process considering a fleet of independent items, the degradation process, and the generation and execution of maintenance plans. The model with different parameters has been used to perform five experiments on the maintenance process. In all the experiments, we have studied three different prioritization policies between PM and CM work on the maintenance plan. Thus, the prioritization policy between the maintenance program and unexpected events. These experiments consider a fleet of independent items with four states and a deterministic degradation process. In the first four experiments, we consider a highly optimized maintenance process, where maintenance is performed just in time before failure. In the last experiment, the number of items is reduced by 10%, resulting in a corresponding increase in the capacity margin.

In the first experiment, we investigate how an imbalance between maintenance programs and the capacity to perform maintenance accumulates a backlog and balances the process. In the two following experiments, we investigate two ways of reducing the PM backlog. First, we increase the PM interval, and secondly, we increase the capacity. Both attempts reduce the backlog. The latter, however, has another effect of increasing the amount of maintenance performed. In the fourth experiment, we study how the system responds to an unexpected event. The results show that a highly optimized maintenance process might be extremely fragile to unexpected events and delays. Although our results show that responding to the unexpected event solves the failure in the short term, it also shows that the solution might cause even more failures to occur. This might lead to the disruption of the maintenance program, transforming a preventive program into a corrective reality. Lastly, in experiment five, we investigate how inaccurate prioritization of PM activities might affect the maintenance process. The experimental results show that inaccurate prioritization might lead to substantial risk exposure. The experiment also shows how the PM backlog might appear to be in equilibrium, where the amount of work created is exactly the amount removed. This, however, as our experiment shows, does not necessarily indicate a well-functioning PM program but is due to the data management policy, where PM activities are removed upon failures. Further, it shows that although the system is in a corrective reality, PM activities are still planned and rescheduled until failure of the item, resulting in unnecessary work.

We have discovered several issues we would like to address in future works. First, a modeling methodology for multi-perspective modeling and simulation tool should be developed to help better understand industrial problems and make risk-informed decisions. Secondly, for the further development of a the tool, the system boundary of the tool should be expanded to include other influential parts to the maintenance process, such as the availability of spare parts and tools, imperfect maintenance, and item availability and criticality, as well as soft variables such as motivation, stress, and competence. Investigations should be conducted on how maintenance organizations respond to resource shortages, such as workarounds. Further on, how resource shortage might affect the data quality in information systems, and how this again affects the resource utilization. Investigations and model development should explicitly model assumptions, delays, objectives, bounded rationality behind decisions, and the relationship and interactions between parts of the system. Additionally, the further development of the tool and methodology would benefit from being developed in collaboration with industrial companies. A collaboration would improve the development to include the factors affecting their specific problems and ensure the usefulness of such tools and methodologies to the industry.

References

Alabdulkarim, A. A., Ball, P. D., and Tiwari, A. (2013). "Applications of simulation in maintenance research". eng. In: *World Journal of Modelling and Simulation* 9.1, pp. 14–37.

Alrabghi, A. and Tiwari, A. (2015). "State of the art in simulation-based optimisation for maintenance systems". eng. In: *Computers & industrial engineering* 82, pp. 167–182.

— (2016). "A novel approach for modelling complex maintenance systems using discrete event simulation". eng. In: *Reliability engineering & system safety* 154, pp. 160–170.

Bengtsson, M. and Salonen, A. (June 2009). "On the need for research on holistic maintenance". In: *Proceeding of 22nd International Congress of COMADEM*, pp. 165–172.

Bokrantz, J. et al. (2017). "Maintenance in digitalised manufacturing: Delphi-based scenarios for 2030". In: *International Journal of Production Economics* 191, pp. 154–169.

Brailsford, S., Churilov, L., Dangerfield, B., et al. (2014). *Discrete-event simulation and system dynamics for management decision making*. Wiley Online Library.

Carroll, J. S., Sterman, J., and Marcus, A. (1998). "Playing the maintenance game: How mental models drive organizational decisions". In: *Debating rationality: Nonrational aspects of organizational decision making*, pp. 99–124.

Forrester, J. W. (1961). *Industrial dynamics*. eng. Cambridge, Mass.

Größler, A., Thun, J.-H., and Milling, P. M. (2008). "System Dynamics as a Structural Theory in Operations Management". eng. In: *Production and operations management* 17.3, pp. 373–384.

Günal, M. M. and Pidd, M. (2010). "Discrete event simulation for performance modelling in health care: a review of the literature". In: *Journal of Simulation* 4.1, pp. 42–51.

Hallan, B. et al. (Apr. 2021). *Rapport etter gransking av brann i luftinntak til GTG4 på Hammerfest LNG, Melkøya*. Tech. rep. 001901043. Stavanger, Norway: Petroleumstilsynet.

Jåsund, E. et al. (June 2014). *Tilsyn med styring av vedlikehold på Heimdal Main Platform (HMP)*. Tech. rep. 001036010. Stavanger, Norway: Petroleumstilsynet.

Linnéusson, G. (2018). "Towards strategic development of maintenance and its effects on production performance : A hybrid simulation-based optimization framework". PhD thesis. University of Skövde, The Virtual Systems Research Centre, p. 226.

Linnéusson, G., Ng, A. H., and Aslam, T. (2018). "Towards strategic development of maintenance and its effects on production performance by using system dynamics in the automotive industry". In: *International Journal of Production Economics* 200, pp. 151–169.

Morgan, J. S., Howick, S., and Belton, V. (2017). "A toolkit of designs for mixing discrete event simulation and system dynamics". In: *European Journal of Operational Research* 257.3, pp. 907–918.

Morrison, B. (2015). "The problem with workarounds is that they work: the persistence of resource shortages". In: *Journal of Operations Management* 39, pp. 79–91.

CEN (Feb. 2017). *NS-EN 13306 Maintenance - Maintenance terminology*. Standard. CEN/CENELEC.

Pedersen, T. I., Vatn, J., and Jørgensen, K. A. (2020). "Degradation Modelling of Centrifugal Pumps as Input to Predictive Maintenance". en. In: p. 8.

Pintelon, L. and Parodi-Herz, A. (2008). "Maintenance: an evolutionary perspective". In: *Complex system maintenance handbook*. Springer, pp. 21–48.

Rausand, M. (2014). *Reliability of safety-critical systems: theory and application.* eng. WILEY.

Rauzy, A. B. and Haskins, C. (2019). "Foundations for model-based systems engineering and model-based safety assessment". In: *Systems Engineering* 22.2, pp. 146–155.

Repenning, N. and Sterman, J. (Feb. 2003). "Nobody Ever Gets Credit for Fixing Problems That Never Happened: Creating and Sustaining Process Improvement". In: *Engineering Management Review, IEEE* 30, pp. 64–64.

Robinson, S. and Brailsford, S. (2014). "Discrete-event simulation: A primer". In: *Discrete-event simulation and system dynamics for management decision making*, pp. 10–25.

Rødseth, H. and Schjølberg, P. (2017). "Maintenance backlog for improving integrated planning". eng. In: *Journal of quality in maintenance engineering* 23.2, pp. 195–225.

Sterman, J. (2000). *Business dynamics : systems thinking and modeling for a complex world*. eng. Boston: Irwin McGraw-Hill.

— (2002). "All models are wrong: reflections on becoming a systems scientist". In: *System Dynamics Review: The Journal of the System Dynamics Society* 18.4, pp. 501–531.

— (2010). "Does formal system dynamics training improve people's understanding of accumulation?" In: *System Dynamics Review* 26.4, pp. 316–334.

Thorsen, A. J., Leto, S., and Jåsund, E. (June 2020). *Tilsyn med styring av vedlikehold på Mongstad - oppfølging av tilsyn fra 2019.* Tech. rep. 001902052. Stavanger, Norway: Petroleumstilsynet.

Valerdi, R. and Fernandes, B. (2011). "Underestimation in the "When It Gets Worse Before it Gets Better" Phenomenon in Process Improvement". In: *Improving Complex Systems Today*. Springer, pp. 3–10.

Warren, K. (2005). "Improving strategic management with the fundamental principles of system dynamics". In: *System Dynamics Review: The Journal of the System Dynamics Society* 21.4, pp. 329–350.

NTNU

Norwegian University of
Science and Technology