Towards a predictive maintenance strategy for Passenger Boarding Bridges at the airport

MSc Thesis

Lester Meijs

Delft University of Technology

Towards a predictive maintenance strategy for Passenger Boarding Bridges at the airport

by

L. Meijs

Master thesis

in partial fulfilment of the requirements for the degree of

Master of Science
in Mechanical Engineering

at the Department Maritime and Transport Technology of Faculty Mechanical, Maritime and Materials Engineering of Delft University of Technology

to be defended publicly on Thursday January 25, 2024 at 14:00.

Student number:	4552970	
MSc track:	Multi-Machine Engineering	
Report number:	2023.MME.8887	
Supervisor:	Dr.ir. Y. Pang	
Thesis committee:	Prof. dr. R. Negenborn,	TU Delft, Graduation Committee Chair
	Dr. ir. Y. Pang,	TU Delft, Supervisor
	A. Caspani,	TU Delft, Examination committee
	O. Maan,	Royal Schiphol Group, Company supervisor

Date: 08-01-2024

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Preface

In front of you, you find my master thesis for the Master Mechanical Engineering - Multi-Machine Engineering at the TU Delft. This thesis has been conducted in collaboration with Royal Schiphol Group and VolkerWessels Infrastructure. First of all, I want to express my gratitude to the graduation committee for their guidance and supervision throughout the research process. I would like to thank my daily supervisor Dr. ir. Yusong Pang for the help and guidance during the research. Especially when the progress was slow, his patience and feedback ensured I continued the research and kept the spirit up. I would also like to thank the chair of the thesis committee, Prof. Dr. R. R. Negenborn, for the feedback and suggestions during the meetings.

Secondly, I want to thank Royal Schiphol Group and VolkerWessels Infrastructure for the opportunity to do my research in collaboration with them. Jan van der Lee, Mark, Jan Floris and Bart for their support and help during this research. I would like to thank my company supervisors from Royal Schiphol Group, Nijs, Wouter, and Oscar, for their guidance and feedback during the project. I would like to thank Jasper and Marinke for helping me feel welcome in the Innovation team at the beginning of the research. Danny, for the feedback he gave me in the last part of the research. Last and most importantly, I would like to thank Wesley and Willem for our fun and engaging discussions. I had a wonderful time doing my research at Royal Schiphol Group, primarily thanks to the atmosphere within the Innovation team.

Above all, I would like to express my deepest appreciation to my girlfriend Tasha, whose unconditional support and encouragement, although she had a hard time understanding what I was saying, helped me throughout the past year. At last, I would like to thank my friends and family for their support during this research. Although asked many times, I can now finally say that the end of the research is here. I hope you enjoy reading this report.

Lester Meijs
Rijswijk, January 2024

Summary

With an increase of almost 63 percent in the last decade, Amsterdam Airport Schiphol is expanding the number of passengers using the airport to get to or from their destination. One of the Airport's objectives is on-time flight schedules to facilitate a pleasant flight process. Delays impose significant risks for the airport, even resulting in compensation claims by passengers at the airlines, which is the primary concern of the airport. To ensure optimal flight processes, the main contractor of Schiphol Airport started to condition-monitor one of the critical assets in the turnaround process at the airport, the Passenger Boarding Bridge (PBB). However, until now, improving the reliability of the PBB when in use, based on the monitored data, takes place after the fault has occurred. Which, therefore, does not use the data's possibilities to predict the future health state of the PBB to its maximum.

The PBB is classified as a critical asset in the turnaround process. By installing additional sensors at three PBBs at Schiphol Airport, the goal is to enhance the current maintenance strategy and increase the reliability of the PBBs. At Schiphol, corrective and preventive maintenance are executed. Corrective maintenance is done when a sudden failure of the PBB happens, and a maintenance ticket is sent to the maintenance mechanics of the main contractor, VolkerInfra. Preventive maintenance is planned on a quarterly, half-year and yearly basis. To get more insight into the behavior of the PBB, VolkerInfra installed extra sensors to monitor the bridge's condition in real-time. However, the different failure mechanisms must be made clear to better understand what needs to be predicted. Based on the analysis of the Site Acceptance Test protocol and the maintenance tickets, it was found that the PBB mostly fails due to technical causes instead of degradation failure. Furthermore, the conclusion was drawn that the classifications of the maintenance tickets were too general to derive conclusions from to know what caused the failure. Without the root cause of failure, an accurate prediction model can not be made to predict and prevent the impending failure. It was, therefore, necessary to include finding the root cause of the failure of the different sub-systems in the prediction model.

From the literature, it was found that the development of a predictive maintenance (PdM) strategy for multi-component systems is still in the beginning phase compared to single-component models. As the complexity increases with more components, it is critical to ensure an easy-to-understand solution. Therefore, combinations of single-model approaches are recommended when developing a PdM strategy. Here, the advantages of each model can be used to reduce the system's complexity. Second, the objective of a PdM strategy of a multi-component system could be conflicting within the system, and a trade-off must be found. With the different components within the system, the relationships between the components and predictions of the health uncertainties must be included to describe the system accurately. PdM can be enhanced enormously with the upcoming Industry 4.0 and the nine pillars. Especially in data-driven methods, Big Data and the use of sensors enable the monitoring of a system at all points. However, this translates into enormous data storage and unnecessary data collection. With the complexity and size of multi-component systems, a well-defined architecture is needed to ensure that the correct data is collected. From the Industry 4.0 pillars, the Cyber-Physical Systems (CPS) architecture can be used to develop a PdM strategy.

This CPS architecture is used to develop the PdM strategy for the PBB. With the 5C architecture presented in the literature, a lower and higher-level model is created for a multi-component system. The lower-level model consists of the connection, conversion and cyber layer. In the connection layer, the root causes of failure are first determined before data collection occurs. It was found that Big Data enables accessible data collection but can take enormous proportions in size. Therefore, a clear vision of what data to collect must be present. In the conversion layer, feature extraction and dimension reduction take place. With this data to information part, the health status of each sub-system is determined in the cyber layer and serves as the output of the lower-level model. The higher-level model consists of the cognition layer and the configuration layer. In the cognition layer, the health status of each sub-system is used to predict the system's health status using a Dynamic Bayesian Network.

Lastly, decision-making to find the optimal time to maintain the system and the individual sub-systems that occur in the higher-level model is executed in the configuration layer.

After developing the theoretical CPS architecture, the first step in implementing it on the PBB was done. Due to the complexity of this multi-component system, it was impossible to implement the whole PBB in the proposed architecture due to time constraints. Therefore, only the canopy is analyzed to prove the proposed CPS architecture. The KPIs were defined as reliability, repair time and repair rate to evaluate the benefits of the newly developed maintenance strategy. A base case was defined based on historical data and the current maintenance situation to compare the newly developed maintenance strategy to the current situation.

The base case input was used to determine the optimal maintenance moment for the sub-systems of the canopy individually. After that, the global opportunistic model was used to determine the optimal maintenance group together with the group's optimal maintenance moment. It was shown that the model can determine the optimal maintenance moment for the sub-system individually and for the optimal maintenance group by finding the lowest repair rate while respecting the availability constraints. By implementing the synthetic dataset in the cognition layer, the DBN updates the failure probability of the canopy drive and a new decision-making of the optimal maintenance moment is made. The results show that the higher-level model can update the model's reliability and choose the maintenance moment with the lowest repair rate while respecting the availability constraint. When applying the decision-making model to the synthetic dataset, the result shows that the optimal maintenance moment is shifted to a later bridge use. By determining this optimal maintenance moment, the proactive repair time can be used, which is less than the corrective repair time. This proactive repair time is also investigated in terms of how it influences the optimal maintenance moment and repair rate for the base and synthetic dataset case. The results showed that the height of the proactive repair time can influence the outcome of the optimal time to do maintenance for one of the sub-systems. The influence of the number of mechanics on the repair rate was significant; it was seen that increasing the mechanics from only one to two already decreased the total repair rate but did not change the optimal maintenance moment.
It was concluded that a predictive maintenance strategy could be developed by implementing a CPS architecture, which can benefit the airport's turnaround process. By addressing the root causes of the system's failure, adequate data collection can be done, enabling continuous health monitoring of the bridge, its sub-system and its components. With these predictions, decision-making can occur, allowing proactive maintenance moments at which the repair rate is at its lowest while respecting the availability constraints of aircraft stand. With this, the reliability of the PBB is justified and improved, and unwanted downtime during the turnaround process is prevented.

For further research, it is recommended to focus on applying multi-model approaches and how to implement this on a multi-component system to reduce the complexity and ease its usage in practice. Second, it is recommended to explore further if a DBN is the right tool for the cognition layer, and if so, more research must be done for the PBB and the conditional probabilities. Finally, validating the decision-making model in the configuration layer is recommended before implementing it in the real world.

Nomenclature

AR	Expected Arrival Rate
BN	Bayesian Network
CBM	Condition-based maintenance
CPS	Cyber-Physical System
CRT	Corrective Repair Time
DBN	Dynamic Bayesian Network
$FMECA$	Failure Mode, Effect and Critical Analysis
GDA	Acyclic Directed Graph
IoT	Internet of Things
LLM	Large Language Model
$MTBF$	Mean Time Between Failures
NLP	Natural Language Processing
PBB	Passenger Boarding Bridge
PdM	Predictive maintenance
PHM	Prognostics and Health Management
PLC	Programmable Logic Controller
PRT	Proactive Repair Time
RR	Expected Repair Rate
RUL	Remaining Useful Life
SAT	Site Acceptance Testing
TR	Expected Total Rate

List of Figures

List of Tables

Contents

1 Introduction

The aviation industry has seen enormous growth (International Civil Aviation Organization, 2019). This growth has led to the will of airports to expand in size and flight numbers (Niestadt, 2021). This increase in capacity needs to be captured between the existing limits of the airport. With airports functioning within the top of their limitations, delays impose a risk for the airports. Besides the passengers' complaints and the inconvenience at moments of delays, the delays can induce significant costs for the airlines, the primary concern of the airport company (de Alvear Cardenas et al., 2017). It was estimated that the average cost of aircraft block time, the time difference an aircraft goes into and out of the blocks, also known as the turnaround time, in the United States was 80.52 dollars per minute delay (Airlines for America, 2022). This resulted in an overall cost of 33 billion dollars due to delays in the United States in 2019 (Federal Aviation Administration, 2020). The Passenger Boarding Bridge (PBB), figure 1.1, is a critical asset in the turnaround process. The bridge ensures that passengers can walk dry and in a comfortable climate regardless of the conditions outside.

Figure 1.1: The Passenger Boarding Bridge

Figure 1.2 shows a close-up of the activities happening during the turnaround process. The red blocks indicate the actions of the PBB, the scope of this research, and the blue blocks indicate the activities outside the scope of the research. This figure shows that if the bridge fails after the aircraft goes into the blocks, all activities after and including the passengers' unloading, will be delayed. If the bridge fails after boarding, the aircraft cannot leave the aircraft stand, and the turnaround time will again be delayed.

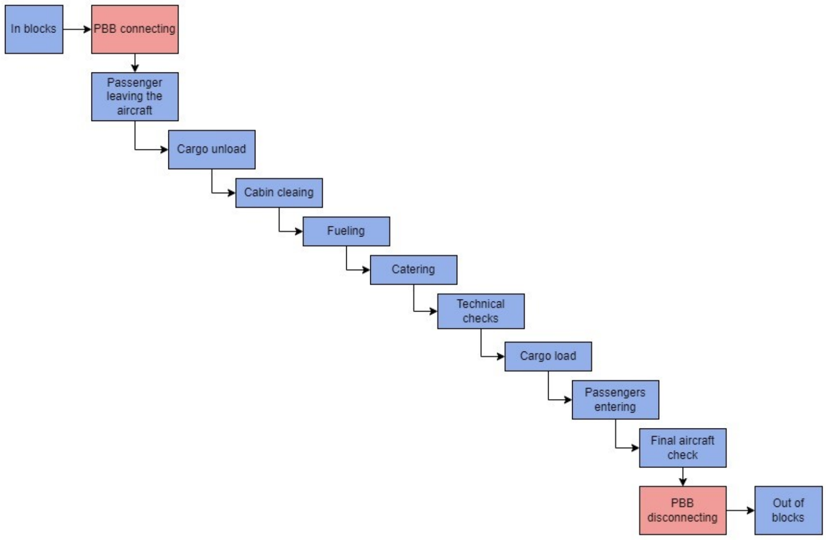


Figure 1.2: The turnaround process

In figure 1.3, all the activities involved at the operational side for one flight process are shown. The high number of relationships between the activities creates a rather complex situation. Different parties are fulfilling these activities around the aircraft stand (ground handlers, cleaning staff, catering etc.) and are involved during the handling of the aircraft, as seen in the figure. This results in that the turnaround process also depends on their willingness to adapt to the situation when the PBB fails.

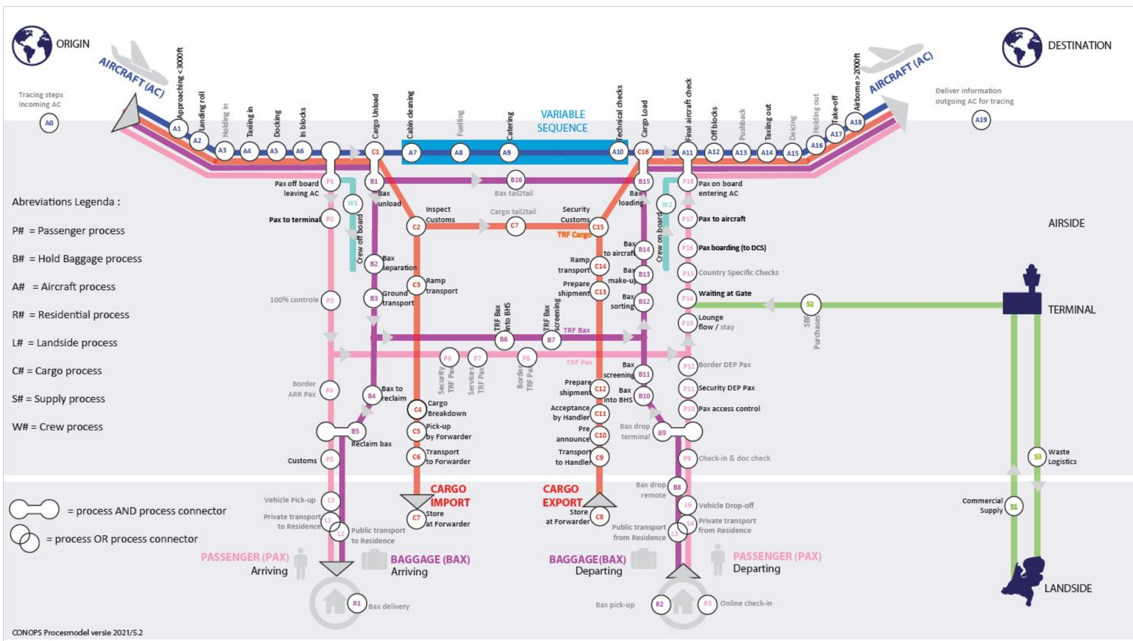


Figure 1.3: Process model of Schiphol Airport

1.1. Schiphol Airport

The PBBs, which are the main subject of this thesis, are located at Schiphol Airport. Royal Schiphol Group is the owner and operator of Amsterdam Airport Schiphol. Schiphol Group also owns and operates Rotterdam The Hague Airport, Lelystad Airport and has a stake in Eindhoven Airport. The airports where Schiphol Group operates create value for society and the economy, where safety is the critical enabler. As one of the oldest international airports, Schiphol landed her first airplane on 19 September 1916. In the following years, Schiphol Airport expanded and became a global hub for departures, arrivals, and transfers (Schiphol, 2022). Today Schiphol aims to grow to be Europe's preferred airport, realizing this by being best in class with respect to asset management. This can only be achieved through collaboration between Schiphol and its contractors (Verheijden, 2018). One of these contractors is VolkerWessels Infrastructuur (VolkerInfra), which focuses on new construction, maintenance, and renovation of Schiphol's infrastructure and systems. As a main contractor of Schiphol, one of their concerns is perceel 2 on Schiphol; see figure 1.4. Within perceel 2, VolkerInfra is crucial in ensuring optimal availability of the infrastructure and all systems needed for managing the airplanes and the aircraft stands, and thus the PBB. (VolkerWessels, 2021).

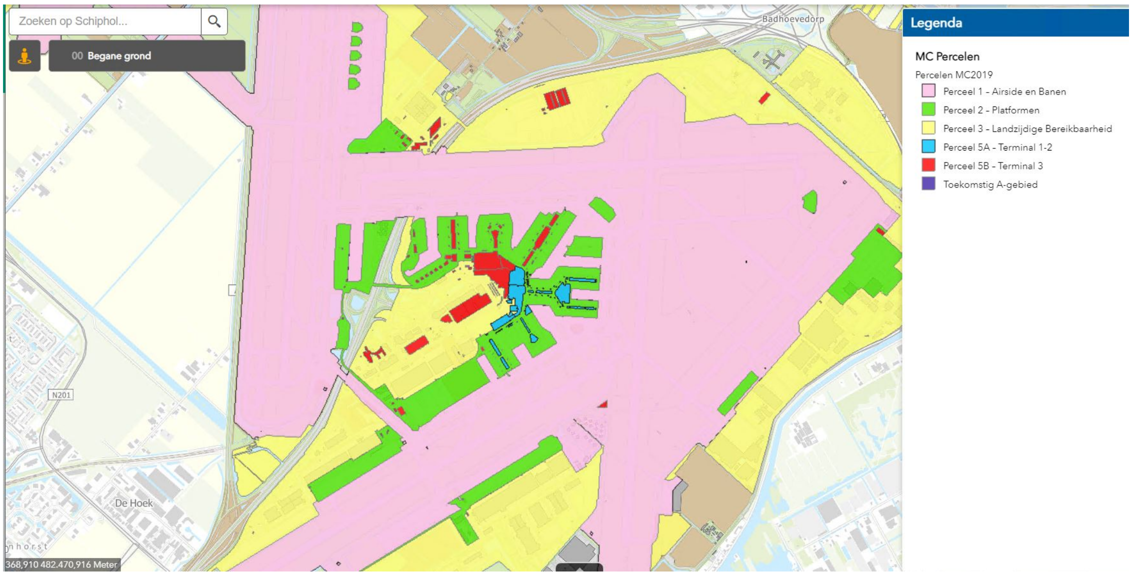


Figure 1.4: Perceel 2 (green) at Schiphol Airport. Snapshot from Schiphol Maps (maps.schiphol.nl)

1.2. Problem definition

The growth of the aviation industry is also visible at Schiphol Airport. From 2010 to 2020, Schiphol Airport increased from 45 million to 72 million passengers using the airport. This is an increase of almost 63 percent more passengers using the airport to get from or to their destination (Centraal Bureau voor de Statistiek, 2022). To facilitate a pleasant pre and after-flight experience, all operations on Schiphol must run smoothly. With the airport functioning 24/7, a tight schedule is maintained, thus limiting space for delays and a last-minute change of assets. Therefore, everything must work correctly to ensure optimal passenger journeys and prevent flight delays. As the introduction mentions, the PBB is critical in ensuring on-time flight processes. If an aircraft is delayed, passengers can claim compensation for the imposed delay by the airline company. Also, a delay can have an influence on the operational processes. A delay can result in the aircraft being unable to depart within its own time slot. Flight time slots are used as timetables for the airport to regulate all the departing and arriving flights. Within the designated time slot, usually 20 minutes (Schiphol Group, 2021), the airlines can use all airport infrastructure (for example, taxi lanes and runways) necessary for the successful operation of the flight (Airport Council International et al., 2020). However, when an aircraft misses its time slot, it must stay on the ground until a new time slot has been found. Within a busy airport like Schiphol, the flight delay will increase even more.

Although all those passengers continuously board flights, the bridges are not continually used. An example of the bridge use can be seen in figure 1.5. The blue bars indicate that aircraft handling activities occur at the aircraft stand. The figure shows that there is significant time between each bridge use, making it attractive to do maintenance in these time slots. The PBBs are not used yearly for more than 500 hours of 3-hour time slots. In these 3-hour time slots, VolkerInfra does preventive maintenance on a time-based schedule (van Barneveld & Verheijden, 2019).

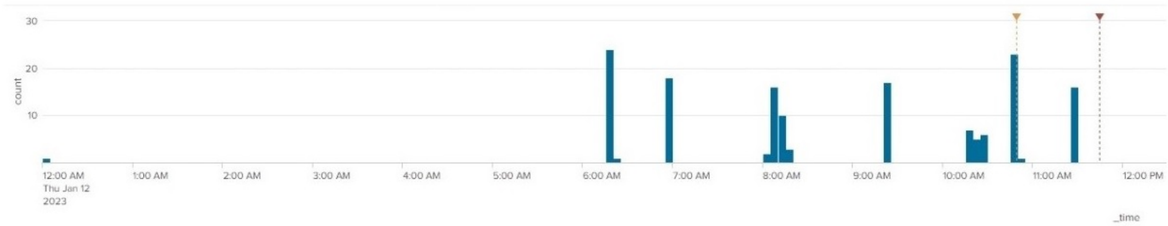


Figure 1.5: An example of the bridge use of stand D16

To get more insight into the failure mechanisms of the PBB, VolkerInfra installed sensors in 2019 on three PBBs at Schiphol, located at stand D16, D18 and D51. Together with the data gathered from the bridge Programmable Logic Controller (PLC), the data is visualized in Volkerinfra's program Flexmonitoring to see if the PBB failure mechanism could be seen or predicted based on the data. An example of visualized sensor data in Flexmonitoring is shown in figure 1.6.

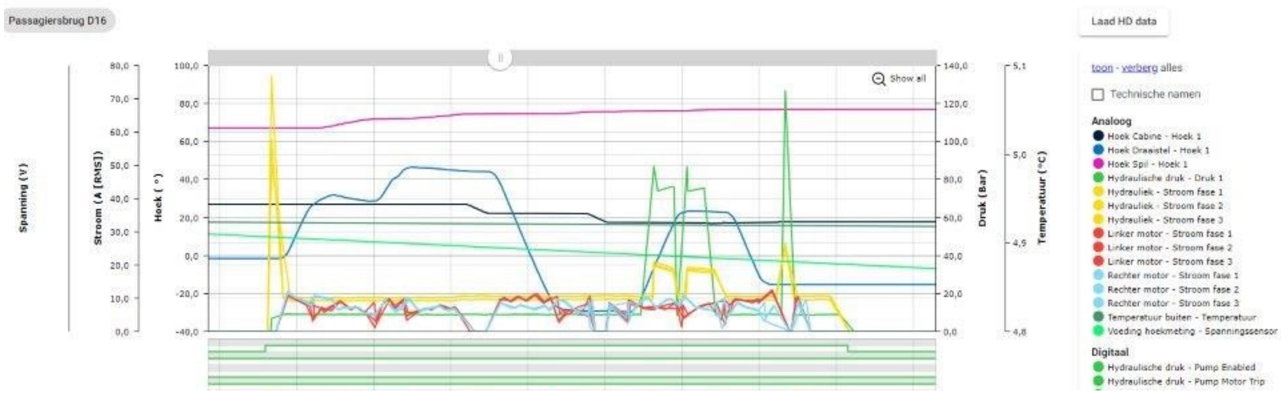


Figure 1.6: Example of the monitored data visualized in FlexMonitoring

If a prediction could be made about when the PBB will be 'out of order' within a specific time range, this would improve the turnaround process on two points:

- First, from an operational-based vision, a prediction model could predict if there is a high probability of failure during the in-time use of the bridge. If this is the case, maintenance activities can be done during the time frame, the unused time slots described above, before the bridge is used. Even if the prediction has a high likelihood of failure during the planned in-action time of the bridge and no maintenance could occur before using the bridge, a gate change, leading to a shift in aircraft stand, could be done. Resulting in no unnecessary downtime and delays.

- Secondly, from a cost perspective. The delay time will be prevented by proactively doing the maintenance described above or changing the gate stand. This will, therefore, reduce the associated cost related to the delay.

However, until now, the monitored data was looked into to see if the bridge's failure could be related to anomalies in the data after the failure occurred. This resulted in that only warnings have been implemented, which will become active if certain thresholds are exceeded. Therefore, the potential of using the sensor data in predicting a failure of the PBB before it occurs has not been reached yet. Secondly, the time the operator uses to (de)connect the PBB varies between 40 seconds and more than 5 minutes. Nevertheless, these are short periods of in-use time of the PBB, resulting in a more complex situation than continuous monitoring of an asset used for hours, which is seen more often in literature. This has led to the following problem statement:

Currently, improving the reliability of the Passenger Boarding Bridge when in use, based on the monitored data, takes place after the fault has occurred. Which, therefore, does not use the data's possibilities to predict the future health state of the Passenger Boarding Bridge to its maximum.

1.2.1. Knowledge gap

The literature is investigated to find how a prediction model can be developed to implement a predictive maintenance strategy for the PBB. Until now, no papers have been found regarding predictive maintenance or other maintenance strategies concerning the PBB. The PBB can be classified as a multi-component system or asset within the airport. When consulting the literature regarding multi-component systems or assets in combination with predictive maintenance, various papers could be found within the production industry, for example, de Pater and Mitici (2021), Dinh et al. (2022), and Rebaiaia and Ait-Kadi (2022), but besides papers on predictive maintenance of a baggage handling systems (Gupta et al., 2023; Koenig et al., 2020; X. Zhang et al., 2022), none regarding an asset in the airport infrastructure. Gashi and Thalmann (2020) state that the research in predictive maintenance of multi-component systems is still in an early phase. According to the authors, research until now is more theoretical than practical, and an investigation into applying theoretical knowledge in practice is needed. Also, a new industry direction, Industry 4.0, is upcoming. The Industry 4.0 pillars can be used within the predictive maintenance strategy, also known as predictive maintenance 4.0, and could add value to the maintenance strategy of the PBB (Carvalho et al., 2019; Shin et al., 2022; Silvestri et al., 2020). This thesis will, therefore, contribute to the literature on the following points:

- Developing a theoretical predictive maintenance strategy for a multi-component system in a practical setting.
- Applying and evaluating a theoretical predictive maintenance strategy in practice based on real-world data.
- Adding knowledge of a predictive maintenance strategy in combination with Industry 4.0 of a multi-component system in a practical setting.

1.3. Research goal and objectives

The research goal has been formulated based on the problem definition described above. The research goal is to develop a prediction model to forecast an impending failure of the PBB to prevent downtime of the PBB during in-time use. This forecast must then result in maintenance activities of the PBB being done proactively, or a real-time gate switching could be suggested. All this together must lead to a decrease in the time delay of the turnaround process of the aircraft, which is currently directly affected if failure of the PBB occurs. The research goal can be divided into the following research objectives:

- The development of a prediction model for the PBB.
- The implementation of the developed prediction model into the current maintenance strategy of the PBB.
- Advice if predictive maintenance is beneficial for the turnaround process of Schiphol Airport and how to expand the strategy to multiple PBBs.

1.4. Research scope

At Schiphol Airport, 91 PBBs are located. 65 of them are CMIC Tianda passenger boarding bridges. In this thesis, the maintenance tickets and related failures of the CMIC Tianda bridges will be used for analysis. These maintenance tickets are from 1 April 2019 to 31 December 2022. The sensor data from the sensors installed by VolkerInfra are monitoring three CMIC Tianda bridges. The passenger boarding bridges are located at stands D16, D18 and D51, marked with yellow markers in figure 1.7. This sensor data is gathered starting from 1 January 2020. Although both datasets were partially collected during the COVID-19 pandemic and the number of flights significantly decreased at Schiphol Airport (Schiphol, 2022), it can be expected that the number of maintenance tickets in regular operation numbers will be higher. However, the data is used for analyzing the failure mechanisms; therefore, the data is assumed to be fit for this research.

An innovation proposal is currently under investigation regarding the bumpers that ensure a safe connection with the aircraft. In this proposal, a new way has been found to decrease the issues when connecting the bumper to the aircraft. If this proposal results in a positive outcome, this new method will be rolled out on the PBBs. It was therefore decided in consultation with the company supervisors to only describe the bumpers but not consider the bumper failures to predict its future state.

Real-time switching of aircraft stands when maintenance of the PBB is not possible can be the solution if a failure of the PBB is predicted. Although an Internet of Things principle could be attractive between the bridges when a shift of gates is needed, no research on the communication between different PBBs will be conducted.

In this research, the objective of the predictive maintenance strategy will, besides predicting the future health state of the system, be to increase the reliability of the PBB in in-time use to prevent a delay in the turnaround process. This means that this research will look into decreasing unplanned maintenance moments. Decreasing the planned maintenance moments, the preventive maintenance intervals, will not be looked into in this research. The PBBs at Schiphol Airport are all but one human-operated. Only maintenance tickets related to the bridge's technical errors will be considered in this research. Human operating errors will, therefore, not be considered in this research.

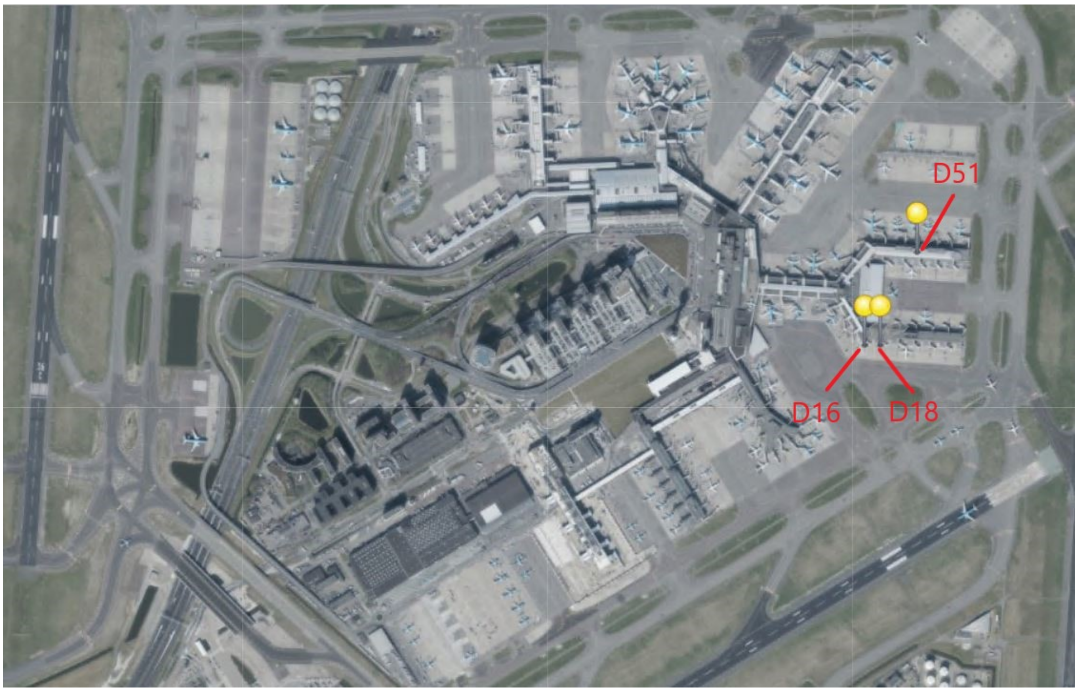


Figure 1.7: Location of stand D16, D18 and D51. Snapshot from Schiphol Maps (maps.schiphol.nl)

1.5. Research questions

To fulfill the research goal and objectives, the following research question has been formulated:

How to realize a predictive maintenance strategy for Passenger Boarding Bridges to benefit the airport's turnaround process?

To answer the research question, multiple sub-questions were composed:

- What are the failure mechanisms of the Passenger Boarding Bridge?
- What is the current state of maintenance activities of the Passenger Boarding Bridges?
- What are the state-of-the-art techniques regarding predictive maintenance?
- How can the prediction model be developed?
- How can the developed prediction model be implemented in the maintenance strategy of the Passenger Boarding Bridge?
- How does the developed predictive maintenance strategy perform in relation to the turnaround process?

1.6. Methodology & thesis outline

To investigate how a prediction model for a PBB can be realized, this chapter gave background information regarding the problem. It defined the reason why this research will be conducted. To visualize the research structure, a conceptual framework is presented in figure 1.8. In chapter 2, the current situation of the PBB at Schiphol will be investigated. In this analysis, the working principle of the PBB will be presented to know how the PBB is operated. From maintenance logs provided by Volkerinfra, qualitative data is analyzed to find the root causes of failure. In the analysis, the maintenance logs will be combined with the Failure Mode, Effect and Critical Analysis (FMECA) and the Site Acceptance Testing (SAT) protocol of the PBB to link the failure mechanism to the sub-systems of the bridge. From here, the bridge's critical sub-systems regarding the operation of the PBB will be defined. In chapter 3, literature research will be executed to investigate what has already been written about predictive maintenance. This will provide a clear overview of predictive maintenance and the knowledge needed about state-of-the-art techniques for predictive maintenance. A Cyber-Physical System (CPS) architecture will be used in this research (J. Lee et al., 2015; Song et al., 2021) and is presented in chapter 4. In the first level, the connection level data will be gathered based on expert knowledge and maintenance ticket analysis. At the conversion level, feature extraction and dimension reduction take place. At the cyber level, the health status of each of the sub-systems is determined. This is done with a multi-model approach. At the cognition level, the system is analyzed to determine the health status of the PBB via a Dynamic Bayesian Network (DBN). Based on the health status of the sub-systems and the system, decisions for actual maintenance activities for the system or the sub-systems are made at the configuration level. To prove the model, a simulation model will be used in chapter 5. Here, a base case will be defined, and after implementing the model and verifying, a simulation will be done to see the performance relative to the current maintenance strategy of the PBB. The results of this simulation will be presented in chapter 6. After the simulation results, the benefits and drawbacks will be discussed in chapter 7, followed by a conclusion in chapter 8. Possibilities concerning further research will be presented in the recommendations.

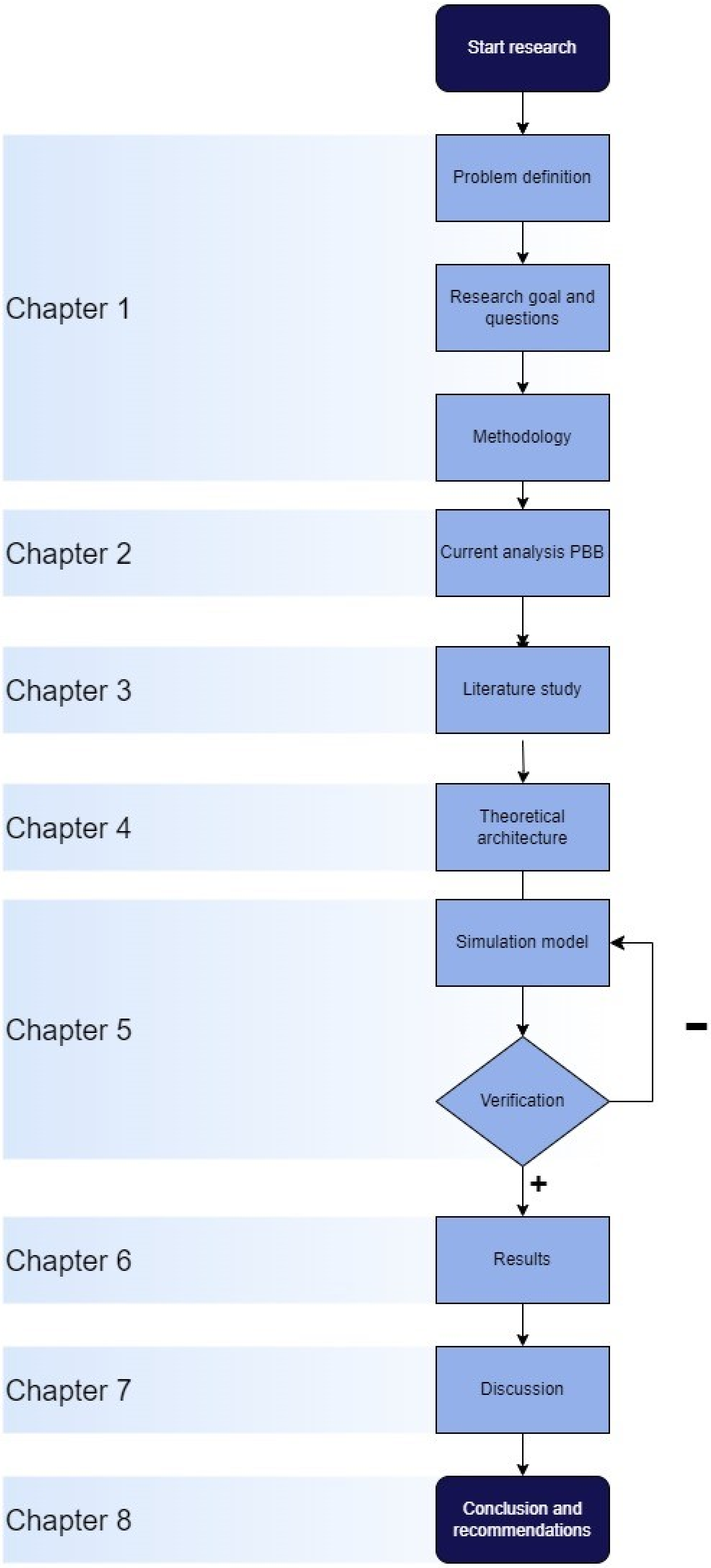


Figure 1.8: Conceptual framework of this research

2 Passenger Boarding Bridge

In this chapter, the PBBs at Schiphol Airport will be discussed. This chapter will answer the sub-questions: ”What are the failure mechanisms of the Passenger Boarding Bridge?” and ”What is the current state of maintenance activities of the Passenger Boarding Bridges?”. First, general information about the PBB and the working principle will be described, followed by the current maintenance procedure in section 2.3. In section 2.4, the failure mechanisms of the different sub-systems of the PBB will be described.

2.1. CMIC-Tianda Passenger Boarding Bridges

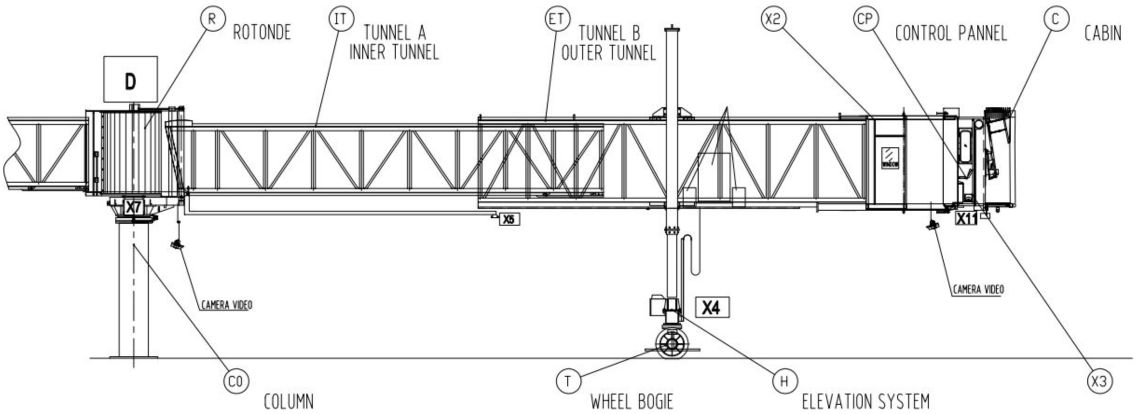


Figure 2.1: Schematic overview of the Passenger Boarding Bridge

The three PBBs are from the company CMIC-Tianda. Figure 2.1 shows a schematic overview of the bridge. The bridge can be divided into different sub-systems. The rotunda is the connection between the bridge and the main structure of the terminal building via a fixed link bridge. When operating the rotunda column, floor, ceiling, and corridor wall panels adjacent to the terminal stay stationary, while the rotunda’s rigid frame and roof rotate on the column. From the rotunda, two telescopic tunnels are connected. As the name suggests, the tunnels have a telescopic crosssection over the length. This is because the tunnels can slide over each other to extend or reduce size when the bridge is operated to reach a specific location. The elevation system supports the outer tunnel. The elevation system enables the bridge to be lowered or raised to achieve the same height as the aircraft doors. Due to the variety of aircraft models and door heights, this height can not be preprogrammed into the bridge as a constant value. The lowering and lifting are done hydraulically. The elevation system is connected to the wheel bogies. The wheel bogie includes the frame, tires, wheel drive reducers and motors and the safety systems for the wheels. The wheel bogie system ensures the bridge can drive towards any location within the workspace. The elevation system and wheel bogies can be seen in figure 2.2.

Figure 2.2: The wheel bogies and the elevation system of the PBB

Figure 2.3: Schematic overview of the aircraft stand

The workspace of the PBB is a limited area within the aircraft stand, which is visualized in figure 2.3. At the end of the telescoping tunnel, the cabin is connected, figure 2.4. This cabin can rotate to the left or right to adjust to the plane. The operator panel is located within the cabin. From this panel, all movements and actions of the bridge are controlled from here. The canopy connects to the cabin, which acts as a roof between the cabin and the plane. Also connected to the cabin is the trim. The trim ensures the bridge automatically levels with the aircraft when height changes. This is because when the aircraft is unloaded, the aircraft will rise when the weight becomes less as the passengers and luggage leave the aircraft (Aviation Learnings Team, 2020).

Figure 2.4: The cabin of the PBB seen from the outside, the canopy and the trim arm

2.2. Working principle

The PBB is used in two situations: from the parking position, the left side in figure 2.5, towards a just landed aircraft, and secondly, in the situation when connected to the plane, the right side in figure 2.5 and it needs to go back towards its parking position. In the following two paragraphs, the PBB procedure in those two situations will be described. It is important to notice that the PBBs are all but one human-operated.

Figure 2.5: The two starting situations of the PBB. On the left is the parking position, and on the right, it is connected to the aircraft.

From starting position towards connection with the aircraft Before any operation with the PBB occurs, the operator must check if the area around the PBB is clear of any obstacles or persons. Only then can the operation of the PBB start to take place. First, a check must be done to see if all the lights of the buttons are working. The PBB is then activated by pressing the button ”HAND”. The button ”HAND” will light up, and the hydraulic pumps will be started. To drive the PBB and connect it to the plane, the following conditions must be met:

- Both canopy switches need to be active; this means that the canopy is not extended
- The trim must be in the upward position
- The ”trim on aircraft” warning must be off
- All three bumper switches cannot be active
- The roller door must be closed

If all conditions stated above are met, the operator can operate the bridge towards the aircraft using a joystick. The possible movements are the rotation of the cabin and the vertical and horizontal displacement of the bridge. These can all be done simultaneously, and the automatic opening of the roller door towards a half-opened stand will take place when the bridge is moved toward the aircraft. When the bridge is 1.5 meters from the plane, the screen ”aircraft detected” notification will pop up, and the bridge will move at a lower speed. The bridge will stop moving forward when one of the three bumper switches is activated. However, to correctly position the bridge with the aircraft, at least two bumper switches must be active. If the aircraft detection and bumper switches are active, the button for the trim lowering will be active. The trim will be lowered on the aircraft, and the notification ”TRIM ON AIRCRAFT” will be visible on the touchscreen. When the trim is active, the roller door will automatically be opened. With the button ”CANOPY DOWN”, the canopy can be connected. Now, the bridge can be used for loading and unloading of passengers.

From connected to the plane back to parking position Also, from this starting point, the surrounding area needs to be checked to be clear of obstacles and persons. The procedure begins with checking the button lights. The green parking button must be pushed and held down within ten seconds. This activates the automatic parking procedure. First, the roller door will close, and the canopy will be moved in. Then, the trim will go back to the upright starting position. The wheel bogies will go towards the same angle as the cabin to ensure the bridge will go backward from the plane in a straight line. The bridge will drive back at low speed until the aircraft is no longer detected, 1.5 meters from the plane. Then, the PBB will go at a normal pace to the preprogrammed parking position. When the bridge is at its parking position, the parking button will light up, and the button can be released. Now, the PBB is ready for the next handling.

2.3. Current maintenance situation

If there is a problem with the PBB, for instance, a malfunction of the bridge, complete failure or a strange noise coming from the bridge, a maintenance mechanic of VolkerInfra can come to inspect the situation. A maintenance ticket needs to be made and sent to the mechanic. The maintenance ticket can originate from three different sources, visualized in figure 2.6. The first way is that a problem occurs with the bridge, and the operator stops the operation and calls business operations. At business operations, the situation is assessed if a mechanic is necessary. If this is the case, a maintenance ticket will be sent to the maintenance interface of VolkerInfra. The maintenance tickets from this interface will be further divided and sent to the assigned mechanics. In the second case, problems with the PBBs are noticed by business operations via Schiphol’s ASCM interface. This interface displays the current status of the bridge; this interface can be seen in figure 2.7. If something is different than supposed, a notification will be made in the ACSM system. Then, business operations must evaluate the report and, if needed, create a maintenance ticket to be sent to VolkerInfra. To relieve business operations of checking the notifications in ACSM, ”smart tickets” are being implemented. These smart tickets directly transfer the bridge’s problems into a maintenance ticket, instantly sent to VolkerInfra. However, the transition of notifications in ACSM to smart tickets is still in progress. When the maintenance mechanic gets the order to inspect the PBB, he or she must be at the PBB within 15 minutes if the handling of the aircraft encounters a disturbance. Otherwise, this time to arrival is 60 minutes.

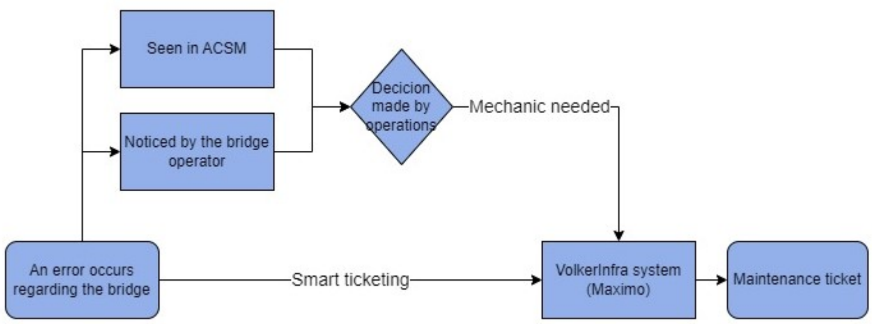
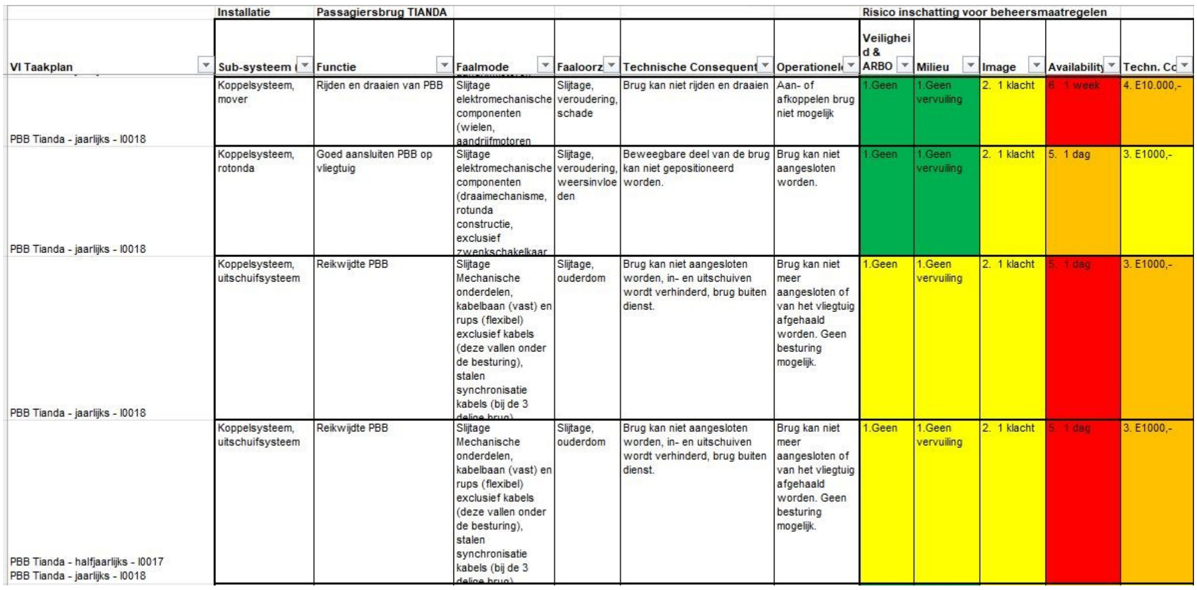


Figure 2.6: Current maintenance ticket situation

Figure 2.7: Snapshot of ACSM for the PBB (in Dutch)

2.3.1. FMECA

Besides the maintenance tickets associated with the malfunction and failure of the bridge, preventive maintenance regarding the PBB is executed. The inspection, actions and the period of the preventive maintenance activities are described in the FMECA. The FMECA *"analyzes and ranks the risk associated with products and process, prioritizes them for remedial action, aiming to reduce their risks and to provide information for making risk management decisions"*(Martins et al., 2018). In figure 2.8, a part of the FMECA document of the PBB is visualized.



	Installatie	Passagiersbrug TIANDA					Risico inschatting voor beheersmaatregelen				
VI Taakplan	Sub-systeem (	Functie	Faalmode	Faaloorz	Technische Consequent	Operationel	Veiligheid & ARBO	Milieu	Image	Availability	Techn. Cc
PBB Tianda - jaarlijks - I0018	Koppelsysteem, mover	Rijden en draaien van PBB	Slijtage elektromechanische componenten (wielen, aandrijfmotoren	Slijtage, veroudering, schade	Brug kan niet rijden en draaien	Aan- of afkoppelen brug niet mogelijk	1.Geen	1.Geen vervuiling	2. 1 klacht	6. 1 week	4. E10.000,-
PBB Tianda - jaarlijks - I0018	Koppelsysteem, rotonda	Goed aansluiten PBB op vliegtuig	Slijtage elektromechanische componenten (draaimechanisme, rotunda constructie, exclusief zwenkschakelkaar	Slijtage, veroudering, weersinvloeden	Beweegbare deel van de brug kan niet gepositioneerd worden.	Brug kan niet aangesloten worden.	1.Geen	1.Geen vervuiling	2. 1 klacht	5. 1 dag	3. E1000,-
PBB Tianda - jaarlijks - I0018	Koppelsysteem, uitschuifsysteem	Reikwijdte PBB	Slijtage Mechanische onderdelen, kabelbaan (vast) en rups (flexibel) exclusief kabels (deze vallen onder de besturing), stalen synchronisatie kabels (bij de 3 delige brug)	Slijtage, ouderdom	Brug kan niet aangesloten worden, in- en uitschuiven wordt verhinderd, brug buiten dienst.	Brug kan niet meer aangesloten of van het vliegtuig afgehaald worden. Geen besturing mogelijk.	1.Geen	1.Geen vervuiling	2. 1 klacht	5. 1 dag	3. E1000,-
PBB Tianda - halfjaarlijks - I0017 PBB Tianda - jaarlijks - I0018	Koppelsysteem, uitschuifsysteem	Reikwijdte PBB	Slijtage Mechanische onderdelen, kabelbaan (vast) en rups (flexibel) exclusief kabels (deze vallen onder de besturing), stalen synchronisatie kabels (bij de 3 delige brug)	Slijtage, ouderdom	Brug kan niet aangesloten worden, in- en uitschuiven wordt verhinderd, brug buiten dienst.	Brug kan niet meer aangesloten of van het vliegtuig afgehaald worden. Geen besturing mogelijk.	1.Geen	1.Geen vervuiling	2. 1 klacht	5. 1 dag	3. E1000,-

Figure 2.8: Snapshot of a part of the FMECA of the PBB (in Dutch)

2.3.2. FlexMonitoring

To get more insight into the behavior of the PBB, VolkerInfra installed extra sensors to monitor the bridge's condition in real-time. This condition-based monitoring gives insight in:

- Three-phase current of the hydraulic motor
- Three-phase current of the left bogie motor
- Three-phase current of the right bogie motor
- The pressure of the hydraulic motor
- The outside temperature
- The cabin angle
- The bogie angle
- Rotonda angle
- Power supply

This real-time monitoring is different from the health status displayed in ACSM. The data coming from the PLC to ACSM is transferred every 3 seconds. The data from the sensors are from a gateway installed in the PLC, sampled at a frequency of 2000 Hertz. However, this is downsampled to 10 samples per second. Secondly, almost all data from the PLC, presented in appendix B, is simply a zero or a one. If a signal needs to be a zero and it is one, the green button in the ACSM interface of the PBB, seen in figure 2.7, will turn red. However, no indication of the cause of this error can be derived. The data from extra sensors installed by VolkerInfra is continuous and visualized in their monitoring program FlexMonitoring, as shown in figure 1.6. FlexMonitoring is developed and owned by VolkerWessels company Asset.Insight. FlexMonitoring has already proved to be crucial in visualizing the monitored data of railroads in the Netherlands. PWC conducted a maturity assessment of PdM on VolkerInfra in 2022, and thanks to the use of FlexMonitoring, it was given a 3.8 on a scale of 0 to 4. This was the reason why FlexMonitoring is also used in the case of the PBB. However, no arguments were documented as to why the specific extra sensors were installed and why other data from the bridge were excluded from the monitoring. Therefore, the next section will analyze the failure mechanisms of the PBB and relate it to the different sub-systems to, in the end, evaluate if the current data is sufficient for a prediction model or if extra data needs to be collected.

2.4. Failure mechanisms

To better understand what needs to be predicted, the different failure mechanisms have to be made clear. Failure of the PBB is here defined as a problem of the PBB that results in a stoppage of aircraft handling. A PBB malfunction can be defined as a situation where the bridge does not function as it should, but no effect will be seen on the aircraft handling. In practice, failure notifications and some malfunction notifications will both automatically be sent to VolkerInfra with a ticket for a mechanic to inspect the PBB and do maintenance if necessary. In this thesis, all maintenance tickets are considered to be related to a failure of the bridge. Furthermore, the data will not include the maintenance tickets classified as operating errors. Although the operational errors result in a maintenance ticket, these have no relationship with actual technical errors regarding the sub-systems and will not be included in this research.

2.4.1. Analyzation of the SAT protocol

The SAT protocol of the PBB is used to check if the PBB fulfills its on-site functions before being put into use after installation by the manufacturers. With different checks, the SAT protocol proves that the various functions and sub-systems of the PBB work as contracted. This SAT protocol includes the situations of the bridge where a deviation is described together with the possible technical causes. From these situations, the deviations that will result in the bridge's failure were extracted. It was concluded that the deviations that will result in a failure of the bridge are caused by seven sub-systems of the bridge: The telescopic tunnels, the elevation system, the wheel bogies, the cabin, the canopy, the trim arm and the roller door. In the next section, the maintenance tickets will be analyzed to see if these sub-systems can be linked to the different causes of the maintenance tickets.

2.4.2. Maintenance tickets analysis

The maintenance tickets of the Tianda PBBs from 1 April 2019 to 31 December 2022 were provided by VolkerInfra. Figure 2.9 shows the number of maintenance tickets per year. As mentioned in 1.4, this research does not consider the maintenance tickets classified as operating errors. Therefore, the numbers in figure 2.9 represent the maintenance tickets, excluding the human error maintenance tickets. This resulted in 1067 maintenance tickets in 2019, 1019 in 2020, 1117 in 2021, 1273 in 2022 and a total of 4476 maintenance tickets for the period 2019-2022.

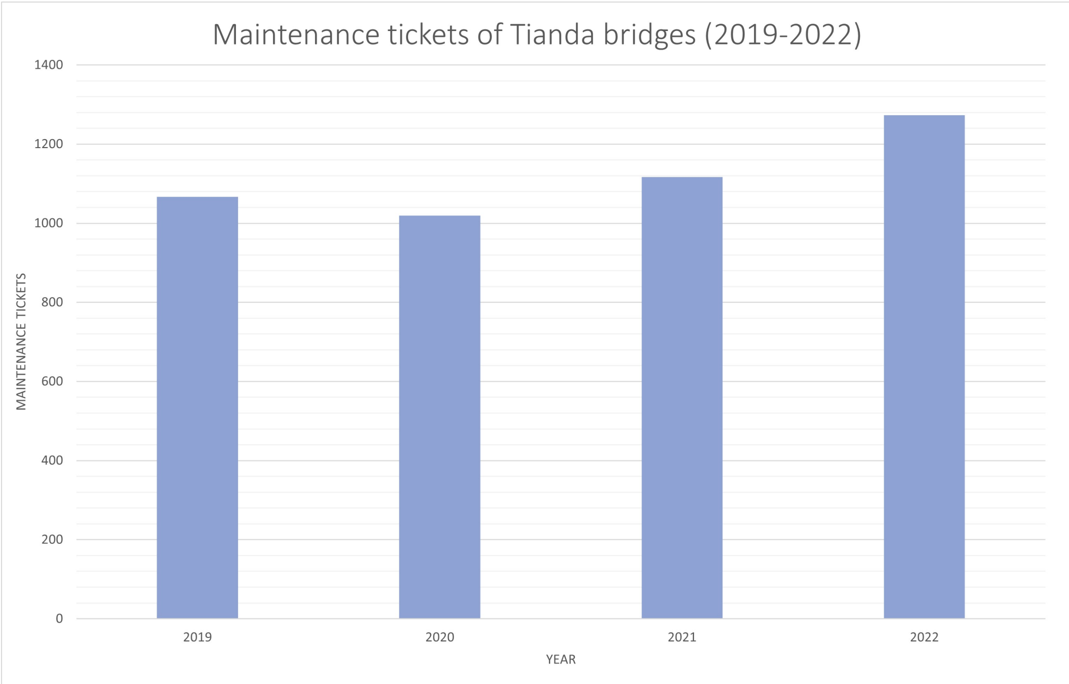


Figure 2.9: Amount of maintenance tickets from 2019 till 2022

The maintenance tickets are classified by VolkerInfra in a fault tree. The first layer of the fault tree is the classification of the maintenance tickets in a specific category. The maintenance tickets can be grouped into the following clusters:

- Bridge (control) does not work
- Bumper failure
- Claxon does not work
- Parking button does not work
- Lifting/lowering failure
- Canopy failure
- Lef/right movement failure
- Parking issue
- Other defects/deviations
- Roller door failure
- Window heating not possible
- Cabin rotation failure

- Driving failure
- Trim failure
- Lights failure
- Video system failure
- Floor heating failure

From figure 2.10 and figure 2.12, the maintenance tickets per category can be seen. Claxon does not work, parking button does not work, window heating not possible, lights failure, floor heating failure and video system failure are not considered in this research due to the low number of tickets present in the last four years compared to the other categories. The bumper failure maintenance tickets are also not further investigated as explained in section 1.4.

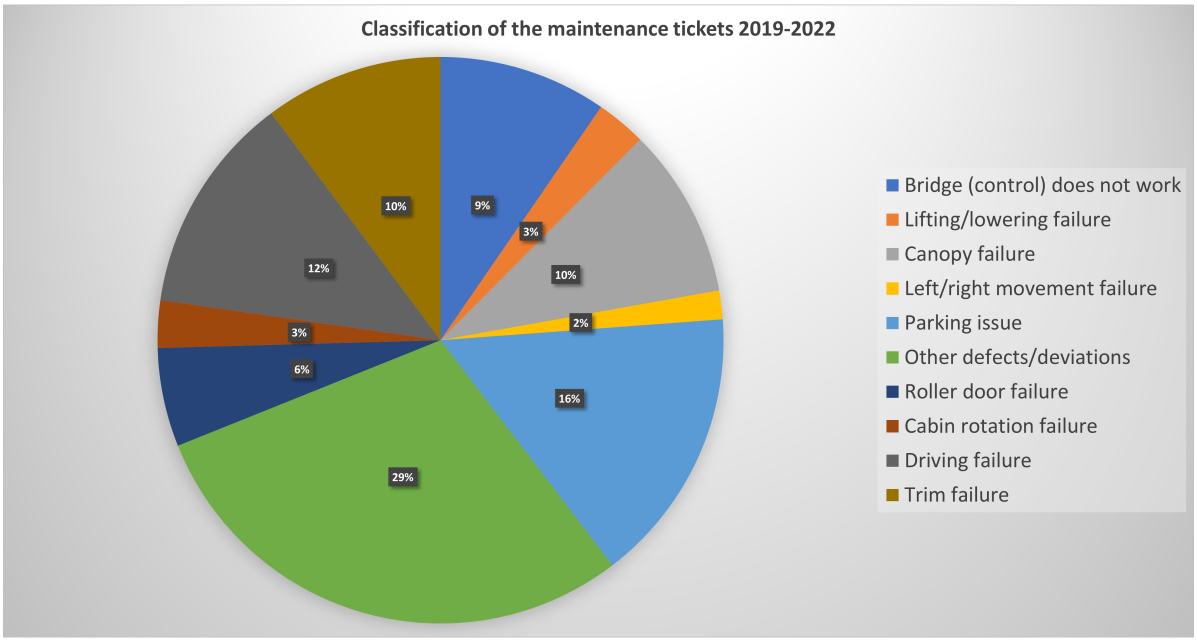


Figure 2.10: Percentages of the maintenance tickets per classification

By clustering the maintenance tickets into their classifications, diagrams are made and presented in appendix C. These diagrams contain the following layers of the fault tree, the causes clustered per component and the general reason for failure. For the classifications: roller door, trim arm, canopy and cabin can be linked to the sub-systems straightforwardly by having the same name. The lifting and lowering classification can be assigned to the elevation system, the left/right and driving classifications to the wheel bogie. For the classification bridge control and parking, a deeper look into the exact reason for failure must be done. In the following paragraphs, the different classifications will be explained and combined with the reasons for failure.

Based on the maintenance tickets for the left/right failures, figure C.5, almost half of the tickets, 30 out of the 72, are related to obstacle detection being activated (42%). These maintenance tickets could be hard to predict without any visual inspection system that checks the surroundings for obstacles. However, when checking figure C.6, it can be seen that a technical failure causes 70 percent of the maintenance tickets. Therefore, further investigation into the actual causes of these tickets needs to be done.

Although the failures related to the cabin have not frequently occurred in the researched period, 117 out of the 4475 maintenance tickets resulted in failure of the PBB due to a failure of the cabin. The limit switch failed 48 percent of the time, followed by the failure while operating and a defect to the drive. In terms of causes, a technical cause was reported for the classification 77 percent of the time.

For the lifting/lowering classification, 124 maintenance tickets were created. From C.3, ten different causes were named for lifting or lowering failure, with failure while operating (30%), valve or relay fault (23%) and tripped switch (25%) being the most occurring faults. From figure C.4, it can be seen that 72 percent of the classifications were related to technical causes in contrast to wear, which is only 2 percent of the causes.

Figure C.9 displays the maintenance tickets associated with the roller door failures. Here no distinct primary cause of failure can be found. Also, for this type of failure, the primary cause of the maintenance ticket is technical failures (73%).

The most frequent classifications for bridge control failure maintenance tickets are failure while operating (48%) and the system jammed (32%). From here, no conclusive answer can be given as to what component the failures can be allocated. Also, the causes, figure C.2, cannot answer this question. Further investigation into these failures is needed.

From 2019 to 2022, a failure related to the canopy resulted in 421 maintenance tickets. In figure C.17, the classification of the maintenance tickets is almost equally split between a defect drive, a closing error, or a limit switch deficiency. The causes, however, figure C.18, are mostly related to a technical origin.

A functional trim arm is crucial to keep the bridge on height with the aircraft and prevent damage to the aircraft. Nevertheless, a failure of the trim arm resulted in 442 maintenance tickets in the research period. In figure C.15, it can be seen that half of the time, the trim arm is at the wrong position, followed by fault while operating 39 percent and a faulty limit switch 8 percent of the time. Unlike the classifications analyzed so far, the cause of maintenance tickets is not entirely dominated by a technical cause. Figure C.16 shows that an external cause for the trim arm causes 26 percent of the classifications.

From the 543 maintenance tickets related to driving failure, figure C.13, 29 percent were classified as a defect of the drive, 28 percent there was a failure while operating and 34 percent of the time the collision protection was activated. Failures related to the limit switch and the emergency button used were less frequent occurring. As said, 1/3 of the time, the ticket was related to the collision protection being activated; however, for 77 percent of all maintenance tickets, figure C.14, the cause was a technical failure. More research is needed on this failure.

With 15 percent of the maintenance tickets classified as parking issue, parking issue is the second most occurring failure after other defects/deviations. For the maintenance tickets related to parking failures, it can be seen that obstacle detection is less represented, only 11 percent. Furthermore, the system jammed 11 percent of the tickets together with a failure while operating 16 percent of the time. The most frequent ticket is a general position alarm. This classification means that somewhere in the procedure, from disconnecting to the parking position, a procedure step took longer than the preprogrammed time threshold.

When looking at the classification of the maintenance tickets, 28 percent, 1270 of the 4475, are classified in the category of other defects/deviations. This category is the most frequently used classification. Investigating the different classifications within other defects/deviations, no specific class could be assigned. It is, therefore, impossible to retrain any information from this class. Hence, also here, more research into the failure mechanism must be done.

2.4.3. Root causes of failure

In this section, the maintenance tickets from the last four years were analyzed. Based on the classifications within the maintenance tickets, the following findings were made. In figure 2.11, the general diagram is portraited. It can be seen that overall, 59 percent of the causes are technical causes compared to 2 percent for wear. A conclusion could be drawn that degradation of the system is not a primary cause of failure, and therefore, prediction models based on life cycles are not of interest to this research. However, this conclusion could be wrong. As explained in section 2.3, preventive maintenance activities are present. It can, therefore, be said that failure due to degradation is prevented due to this maintenance strategy. Nevertheless, the argument that the PBB has a short use interval and, therefore, degradation of the PBB is not going rapidly can also be used. In the end, one of the goals of predictive maintenance is to reduce the preventive maintenance intervals and only do maintenance based on the monitored condition of the PBB. This results in maintenance activities based on degradation that could be present in the predictive maintenance strategy. Thus, a prediction model based on life cycles must be included, either as a separate model or embedded, in the predictive maintenance strategy.

Per sub-system, the reasons for the maintenance ticket were clustered by the fault tree of VolkerInfra. However, most of these classifications were too general to derive conclusions from to know what caused the failure. With the root cause of failure, an accurate prediction model can be made to predict and prevent the impending failure. To overcome this issue, the free text in the maintenance tickets the maintenance logs, can be analyzed. In this free text, the maintenance mechanic can write what the issue was and how the problem was solved. It is, therefore, necessary to see what information these maintenance logs contain to find the root cause of the failure of the different sub-systems. This also means that no evaluation of the monitored data based on the placed sensors can be done, as the root causes of failure are unclear.

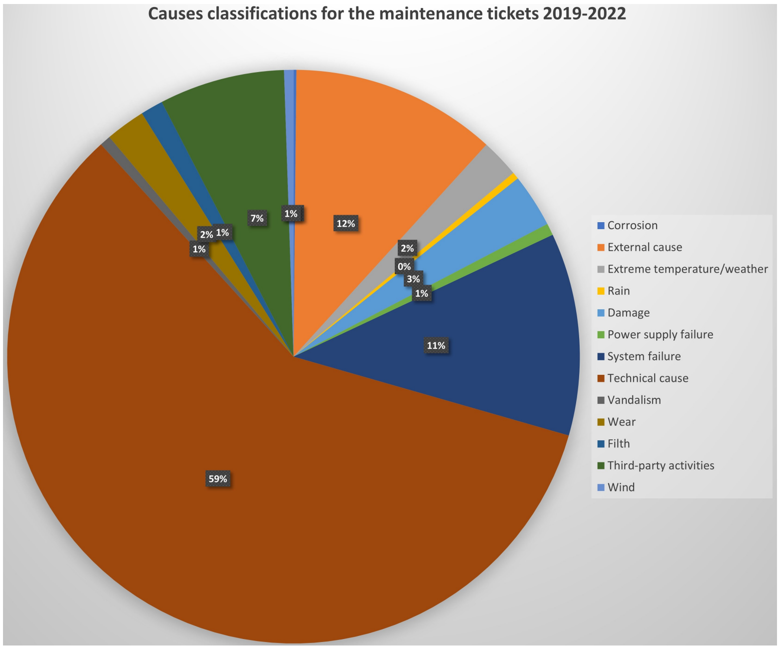


Figure 2.11: Causes classifications of the maintenance tickets 2019-2022

2.5. Conclusion

In this chapter, the research object has been investigated. From analyzing the current state of maintenance activities, it is concluded that reactive and preventive maintenance are conducted now. Sensors were installed at three PBBs to get insight into and overcome the sudden failure of the PBB during in-time use. With these sensors, the data is visualized in VolkerInfra's software package, Flexmonitoring. However, it is unclear how this data will prevent failure mechanisms and which ones. Therefore, the maintenance tickets were analyzed to find the root causes of failure. From this analysis, it was concluded that these tickets were too general to derive any root causes of failure. This results that in the next chapter, literature will investigate what the current state-of-the-art techniques are concerning predictive maintenance and how to develop a prediction model for application in practice.

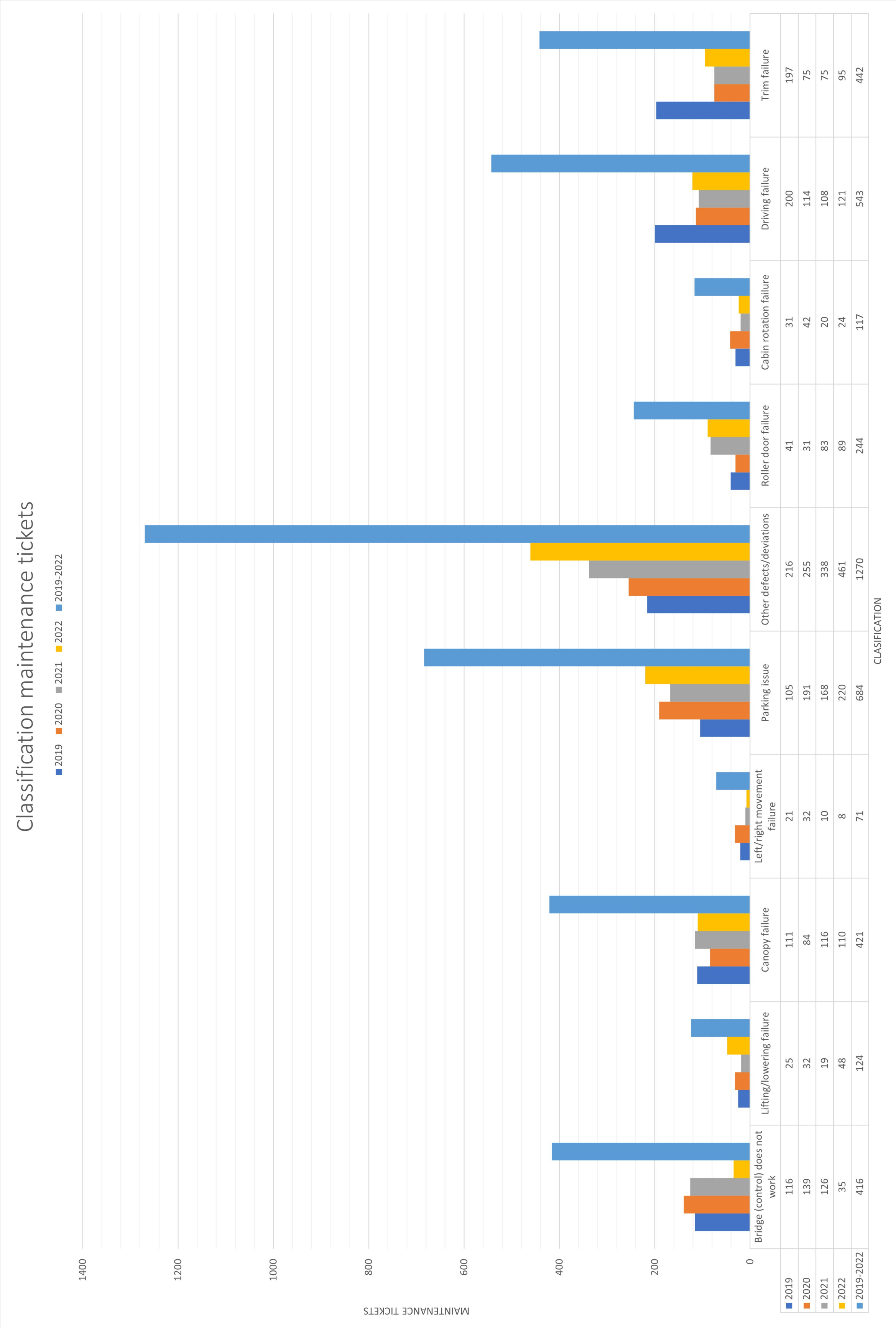


Figure 2.12: Clasiffications of the maintenance tickets

3
Predictive maintenance

In this chapter, a literature survey is presented. In this chapter, the sub-question: ”What are the state-of-the-art techniques regarding predictive maintenance?” is answered. In section 3.1, the different maintenance strategies are briefly explained. In section 3.2, the predictive maintenance approaches are explained, with the objectives in section 3.3. In section 3.4, the usage of Industry 4.0 in predictive maintenance is introduced. At last, the combination of Industry 4.0 and predictive maintenance for multi-component systems is presented in section 3.5.

3.1. Maintenance strategies

Maintenance can be described 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”* (Schenkelberg et al., 2020). During the years, maintenance was seen as an additional manner executed when the system failed, also called reactive or corrective maintenance. As reactive maintenance was costly and resulted in unwanted downtime of the system, maintenance gained interest as it could help improve the performance and reliability of the system, and preventive maintenance was introduced. Instead of doing maintenance after a failure, maintenance on a time-scheduled basis was executed. From preventive maintenance, predictive maintenance arises as maintenance is planned on the system’s condition. To in the end, not only predict the failure but also control and solve the problem, predictive maintenance evolved into prescriptive maintenance (Achouch et al., 2022; Nemeth et al., 2018). In the following subsections, the different maintenance strategies are explained.

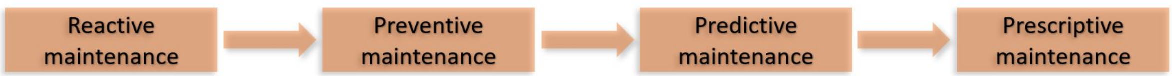


Figure 3.1: The evolution of the different maintenance strategies

3.1.1. Reactive maintenance

The oldest maintenance strategy is reactive maintenance. In reactive maintenance, the maintenance happens when the fault is detected or when the error is notified (Zonta et al., 2020). With this run-to-failure method, a company can use its products to their maximum and, as a result, have maximum availability. Financially, this method could be attractive because the company does not spend any money on maintenance until a breakdown occurs. However, this does not mean that the cost of repairing or replacing the component would be less than applying maintenance along the way. Secondly, spare parts should be available and in stock for critical components if a quick reaction to failure is needed. Otherwise, the reliance on immediate delivery of spare parts is required, and the product will encounter a downtime, which in the end has its influence on the availability of the system in the long term (Ran et al., 2019; Shukla et al., 2022).

3.1.2. Preventive maintenance

Preventive maintenance schedules maintenance at specific points in the future, also known as time-based maintenance. The frequency of these time points or slots is determined by experience or statistical characteristics of the equipment. The bathtub curve is a frequently used graph for equipment failure, presented in figure 3.2. In this curve, the equipment's failure probability is assumed to be the highest at the start of the lifetime, for example, after installation. After the start-up period, the probability of failure will even out in the normal life period. After the normal life period, the chance of failure will increase again. Preventive maintenance thus schedules maintenance activities always to maximize the availability of the equipment based on the data of the lifetime. However, these planned activities could be unnecessary because no problems could be discovered, and unwanted equipment downtime is happening. Also, the chance of critical failures is present due to failure before planned maintenance with unplanned downtime and possibly high cost of repairs (Ran et al., 2019; Selcuk, 2017; Shukla et al., 2022).

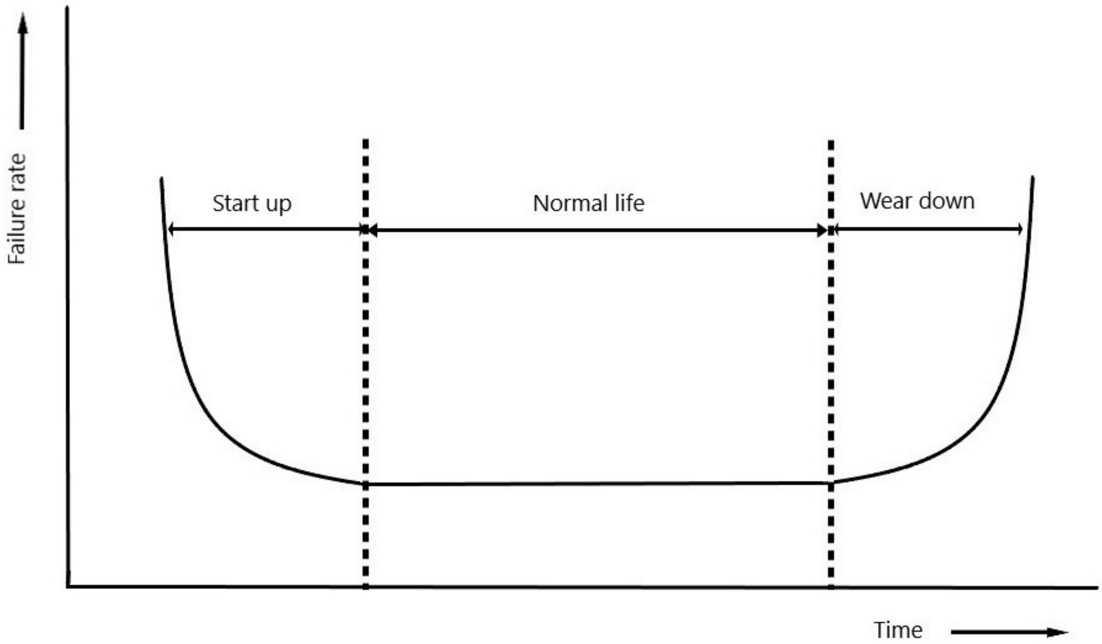


Figure 3.2: Example of a bathtub curve of a piece of equipment. Figure based on Ran et al. (2019) and Shukla et al. (2022)

3.1.3. Predictive maintenance

The next phase in enhancing the maintenance strategy after preventive maintenance is predictive maintenance (PdM). PdM relies on the continuous monitoring of the actual status of the system. Based on the data collected, a prediction will be made of the remaining time the component or the system will be in function or is likely to fail. Then, a trade-off will be made between the condition of the component and the maintenance frequency (Ran et al., 2019; W. Zhang et al., 2019). This proactive method will, instead of time-based maintenance, only plan maintenance when needed and reduce unnecessary downtime due to on-time failure detection (Carvalho et al., 2019; van Dinter et al., 2022). Within the literature, the following terms can be found as different, overlapping, or even a result of each other within the context of PdM. However, the terms will be explained separately to have a clear definition of the terms when continuing with the research.

Condition-based maintenance

Condition-based maintenance (CBM) based its decisions on repair activities solely on the current condition state of the system. Monitoring the system's condition consists of a maintenance program of three steps: data collection, data processing and maintenance decision-making. CBM can be used to activate alarms before a fault in the system occurs. By gathering continuous or periodic information about the state of the system, the CBM model can monitor the health of the system and intervene with repair activities before the system fails (Achouch et al., 2022).

Prognostics and Health Management
While CBM emerged and developed within the industry, Prognostics and Health Management (PHM) originated in military aviation. Where CBM monitors the system condition, PHM tries to go one step further and tries to manage the health of the system. However, this could include CBM as a monitoring technique. PHM allows one to act proactively and take the needed steps to avoid system failure (Tinga & Loendersloot, 2014). Achouch et al. (2022) describe PHM as "the ability to assess the development of future degradation or errors of a system: it guarantees the operation of the system and is an important step before being able to describe the different maintenance scenarios used for the prediction and prevention of malfunctions."

Remaining Useful Life
The Remaining Useful Life (RUL) is defined as the period that a component is alleged to be functioning before maintenance activity needs to take place. This period can be in days, weeks, hours, operating cycles or any other quantity. Although the prediction can provide early warnings of the failure of the component, this prediction depends on the accuracy of the predicted RUL. This results in that the RUL needs to be constructed together with a confidence measure to show the degree of certainty of the RUL (Achouch et al., 2022).

To achieve a PdM strategy, the following steps need to be followed (Achouch et al., 2022; Selcuk, 2017):

First, the researched object needs to be understood. This means knowing how the system operates, why the object fails, what is already measured, and what the goal of the strategy is. For effectively applying the strategy, the root cause of failure must be known instead of only addressing the symptoms of failure. Second, data must be gathered, understood, and prepared for further use. Here, understanding the object and knowing the root cause play a role, as gathering data about conditions that are not used in the end should be avoided. Third, the data must be used in a model for predicting the upcoming failure or the future health state of the object. Fourth, the model's accuracy must be known to assess how the model will perform. After that, the model should be deployed in the researched situation. If the model is not first evaluated correctly, it will fail to give a proper description of the real world. In this step, it also means that the steps explained above are executed properly and/or addressed otherwise in the evaluation before deploying the model. The fifth, also the last step, is decision-making. Based on the PdM goal, decision-making takes place based on the outcome of the used model.

3.1.4. Prescriptive maintenance

Prescriptive maintenance is the final phase in enhancing the maintenance strategy. Prescriptive maintenance not only forecasts the failure in the system but also uses state-of-the-art technologies, for instance, Artificial Intelligence, to asses the cause of the problem and provide a framework to improve and optimize the maintenance processes (Shukla et al., 2022; van Dinter et al., 2022). It aims to control the problem instead of only predicting the occurrence (Nemeth et al., 2018). However, in this thesis, prescriptive maintenance will not be investigated further.

3.2. Predictive maintenance approaches

Within the literature, three approaches for a prediction model can be classified, as well as a combination of the three approaches, also known as a multi-model approach (Montero Jimenez et al., 2020). The first approach is the physical-based approach, which relies on the mathematical modeling of the system based on physical laws. The second approach is the knowledge-based approach, where the model is based on the stored rules and expertise of the system (Cao et al., 2022). The last approach is the data-driven approach; this approach is currently upcoming and gaining interest because these models are implementing artificial intelligence in the form of machine learning to find patterns and anomalies to predict the future state of the system (Shafiq et al., 2022). In the following three sections, the three approaches are explained in more detail.

3.2.1. Physical model based

The physical model relies upon the full understanding of the system translated into the physical, electrical, chemical or mechanical stresses which act on the system and result in failure. This approach uses knowledge about the physical behavior of the system and uses physical laws to predict the degradation of the system or its components (Kwon et al., 2016). The physical behavior is translated into a mathematical representation of the system and reflects how the system responds to the different stresses forced upon the system or its components. This can be on a macroscopic level or a microscopic level. To formulate the physical model and obtain an accurate description of the system, it is vital to understand the failure mechanisms of the system and relate them to the physical laws. This results in an accurate and good estimation of the RUL. However, if the failure mechanism of the system can only be partially understood or not fully linked to the physical laws, the prediction model will become more unreliable (Cao et al., 2022).

3.2.2. Knowledge based

A knowledge-based prediction model relies on a knowledge base that stores the experience regarding the system. This experience can be represented as rules, cases or facts about the system. This experience can then be used to forecast failures or degradation of the system. Where the current situation is compared with the data of an earlier situation, actions will be executed if necessary (Cao et al., 2022). Knowledge-based approaches can be further divided into three subclasses (Cao et al., 2022; Montero Jimenez et al., 2020); the first class is the case-based approach. In this approach, situations are stored as cases. If a new problem arises, the most similar case is chosen from the stored database, and the solution connected to the stored case is reused for the faced problem. If this solution also solves the newly encountered problem, the database is updated with this new problem solution case. However, finding the right characteristics to describe the cases can be challenging (Montero Jimenez et al., 2020). Second is the rule-based approach. The rule-based approach performs based on expert knowledge translated into a pre-determined set of rules, also known as an Expert System. Domain knowledge of the system is extracted from domain experts in a set of situation-action rules. The rules are expressed in the form of IF-THEN. IF a condition, mostly a fact, THEN a consequence is happening. This consequence will affect the situation of the system (Cao et al., 2022; Ran et al., 2019). The third is the fuzzy knowledge-based approach, and these models use the same structure as the rule-based system. Where the rule-based system uses boolean logic, so only a true or false proposition, fuzzy logic can use intermediate values to describe the truth or falsehood of a proposition (Montero Jimenez et al., 2020). For knowledge-based models, acquiring accurate knowledge from experience can limit the prognostic part of the predictions. Furthermore, another drawback can be obtaining access to experts or sources to share knowledge about the systems. Also, dealing with new situations and predicting them based on previous knowledge can affect the system's reliability. However, knowledge-based models are beneficial when explaining the steps taken. Using strict rules and expert knowledge, the reasoning steps taken in this model can be explained and justified (Montero Jimenez et al., 2020; Ran et al., 2019).

3.2.3. Data-driven

The third PdM approach is a data-driven model. Data-driven approaches use data analytics and machine learning to predict the system's future state and detect anomalies in the data regarding the system. This approach uses internal and/or external covariates to predict the system's state. Sensors present or placed within the system are used for internal covariates, such as vibration or current, and are only active when the system operates. External covariates are present even if the system is not running, for example, weather data (Kwon et al., 2016). Data-driven approaches can be further classified into the following subclasses: statistical, stochastic and machine learning. Statistical models are used in PdM for analyzing the degradation of the system and predicting the remaining life of the system. This degradation analysis compares the system's behavior with the known behavior based on the collected data (Montero Jimenez et al., 2020). Frequently used statistical models in PdM are regression analysis, autoregressive models, and Bayesian models. An overview of identified applications can be found in Montero Jimenez et al. (2020). Stochastic models are the second classification of a data-driven approach. As the name already suggests, stochastic models use stochastic processes as the fundamentals of the prediction model (Montero Jimenez et al., 2020). From literature Montero Jimenez et al. (2020) identify three stochastic diagnostics processes: Gaussian, Markov, and Levy.

The potential advantages, drawbacks and related applications can be found in Montero Jimenez et al. (2020). The authors state that stochastic models are generally highly suitable for degradation modeling based on their regression capabilities. However, this is paired with high computational power, uncertainty management and the need for advanced mathematical knowledge. The third data-driven classification is the model incorporating machine learning. This classification uses data analytics in combination with machine learning algorithms to detect anomalies in the data and make a prediction based on the internal and external covariates (Kwon et al., 2016).

3.2.4. Multi-model approaches

The approaches stated above can be seen as single-model approaches. For a complex system, these models only partly address the diagnostic and prognostic task of the system (Montero Jimenez et al., 2020). Montero Jimenez et al. (2020) state that based on their literature study, the reviewed studies mostly propose models to overcome the weak points of the system. With a complex system, the potential faults and failure modes will increase, as well as the number of data, resulting in more prognostic and diagnostic tasks for the model made. A multi-model approach could be implemented to overcome the complexity of the system. In figure 3.3, combinations of the approaches can be seen. Combinations between knowledge-based models are not often used in literature. The combination still has the same issues and challenges as the single-model approach. However, combining knowledge-based models can reduce the complexity of the system through reasoning. With multiple data-driven models, neural networks are the most used. Multiple physics-based combinations can be implemented to use the law of physics to increase the accuracy of the output. However, the combination of multiple physics-based models is not often seen; the reason for this is that implementing these models will require a high amount of knowledge of the system in terms of physics, mathematics and the technical components of the system. When combining knowledge-based models with data-driven models, the advantages of the two models are combined. Resulting in the combination of the knowledge of human experts with the diagnostic and prognostic strength of the data-driven model. This can help in analyzing heterogeneous data sets coming from the system. Increasing the accuracy of a physics-based model can be achieved by combining it with a knowledge-based model. Upcoming and most common in research is combining a data-driven model with a physical-based model. The last combination for a multi-model approach is the combination of the three single-model approaches. Here, the multi-model approach can benefit from the strong points of every model. However, as it combines three models, it also increases the complexity of this approach (Montero Jimenez et al., 2020).

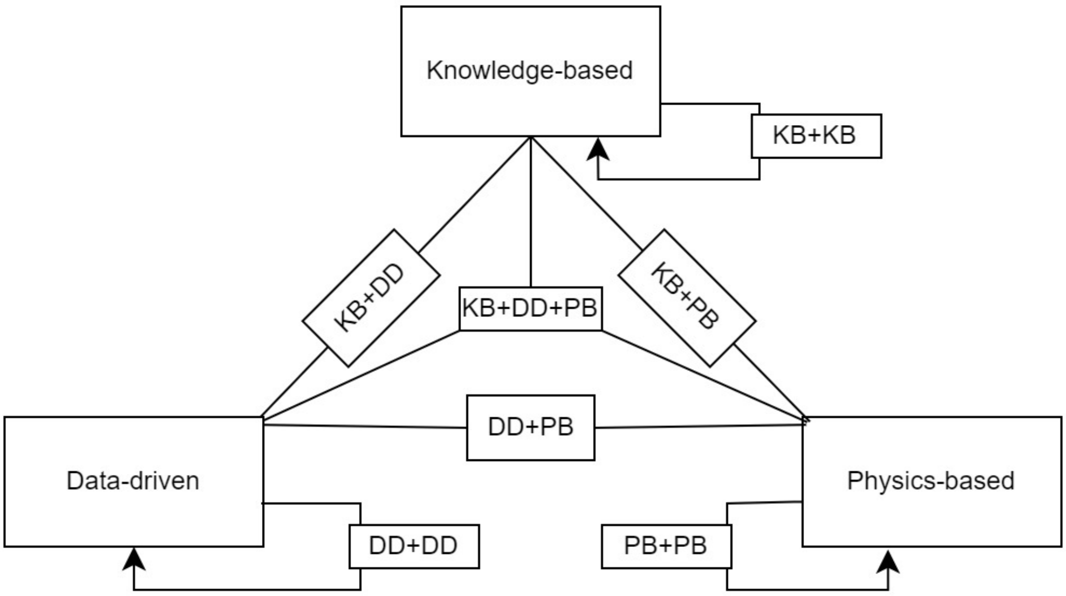


Figure 3.3: The possible combinations of single-model approaches, recreated from Montero Jimenez et al. (2020)

Besides the combinations that can be made between the different approaches, the architecture of the multi-model approach is of importance. Figure 3.4 visualizes the possible architectures. The models can be in series, parallel to each other, or a model can be embedded in another model.

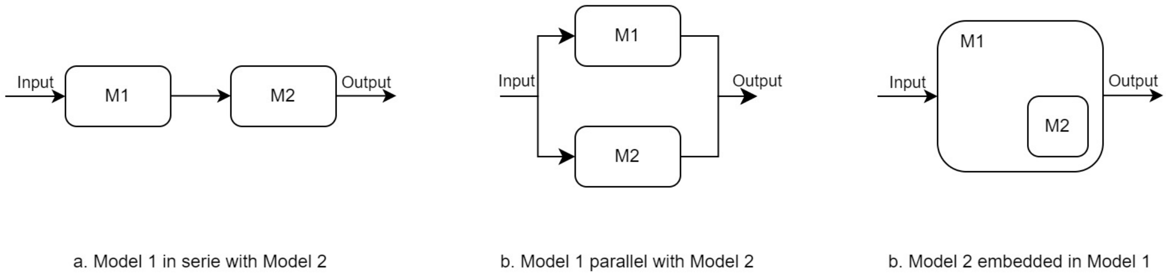


Figure 3.4: Different architectures of a multi-model approach, recreated from Montero Jimenez et al. (2020)

3.3. Predictive maintenance objectives

Although the primary goal of PdM is to predict the system's future state and prevent unwanted downtime, the optimization criteria can be different (Ran et al., 2019). Different optimization criteria can be classified, such as cost minimization and availability, reliability optimization, or feasibility. In Ran et al. (2019), the literature regarding optimizing PdM strategies is reviewed. From the paper, multi-objective optimization can be of interest. From the reviewed articles, it could be seen that only one optimization criterion is usually used as a single-objective optimization approach. However, as the authors state, in the case of a multi-component system, this single objective could conflict with reality. For example, the case of minimum cost objective is given. When aiming for a low cost, this could result in the component's reliability being too low to be acceptable. Therefore, they propose to use a multi-objective approach in this case. A multi-component objective optimization problem tries to find the optimum decision variables to minimize or maximize the different objective functions. However, in practice, finding an optimum for all decision variables is often impossible. Therefore, a trade-off must be made between the different optimization objectives values.

3.4. Predictive maintenance & Industry 4.0

As steam power was introduced for factories, the mechanization of the production processes formed Industry 1.0. Industry 2.0 was realized by mass production systems, followed by developments in information technology and the use of computers for partial automation for Industry 3.0. With the arrival of the internet and the rapid developments in technological innovations, a new era for the industry, Industry 4.0, was arriving. First used in 2011 by the German government, Industry 4.0 is described by the further developments of automation and information technologies (Cannavacciuolo et al., 2023). With Industry 4.0, the physical world and cyberspace are not only connected but synchronized by a digital model of the physical world. This leads to an environment where intelligent supervision and autonomous decision-making processes can enhance the industry (Lesch et al., 2023).

From literature Silvestri et al. (2020) conclude that nine technical pillars drive Industry 4.0: Industrial Internet of Things, Big Data and Analytics, Horizontal and vertical system integration, Simulation, Cloud computing, Augmented Reality, Autonomous Robots, Additive manufacturing and Cyber Security. The pillars are further elaborated in Silvestri et al. (2020). The Internet of Things (IoT) first appeared in 1997. Over the years, IoT evolved in networks where various distributed assets can be connected (Kwon et al., 2016). Industrial IoT further develops IoT by enabling factories to realize machine-to-machine interaction without the intervention of humans. It realizes supply chains to be connected by the internet using sensors and therefore having interconnections between the different physical objects (Silvestri et al., 2020). Silvestri et al. (2020) state that according to the literature, IoT is the basis for CPS. CPS enables the connection of physical objects with the virtual world and removes the boundaries between them. Implementing the Industry 4.0 pillars in the maintenance activities realizes Maintenance 4.0. By implementing and using these intelligent innovations to enhance the current situation, Maintenance

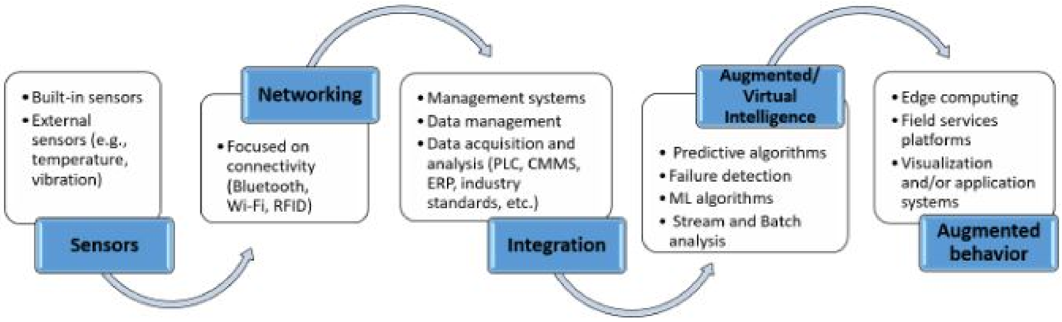


Figure 3.5: Technologies for enabling PdM 4.0 (Werbińska-Wojciechowska & Winiarska, 2023)

4.0 aims to maximize the in-use time of the system (Silvestri et al., 2020; Werbińska-Wojciechowska & Winiarska, 2023). According to the authors Werbińska-Wojciechowska and Winiarska (2023) *"Maintenance 4.0 encompasses a holistic view of data sources, how they are combined, collected, analyzed, and are recommended actions to provide digital support to the function (reliability) and value (management) of assets. As a result, a holistic approach enables effective plant-wide communication between machine operators, maintenance and engineering teams, and management, allowing informed decisions and better utilization of resources. In addition, implementing a holistic approach to PdM provides that individual components are assessed for their value in the entire production chain, and sensors are applied accordingly."* In figure 3.5, the technologies are presented that ensure the translation of PdM to the combination of PdM and Industry 4.0, PdM 4.0. CPS can be used to integrate the Industry 4.0 pillars within PdM and link the physical object to cyberspace, which supports the direction of creating PdM 4.0.

3.4.1. Big Data

Big data in Industry 4.0 is illustrated by the amount of data collected in quantity and variety of data. Big data can be characterized by the five V's, visualized in figure 3.6: Volume, variety, velocity, value and validity. Based on this enormous amount of data, usually with heterogeneous sources, this data is unworkable if directly combined to do a diagnosis or prognostic of the system. It is, therefore, key to managing and processing these large data streams. This can be done by acquiring the knowledge needed to develop the data analysis algorithms. With these analytics and technologies, the comprehension, analysis and real-time decision-making based on the data can improve the PdM strategy and result in a more flexible and reliable monitored system (Achouch et al., 2022; Biard & Nour, 2021; Fasuludeen et al., 2021; Kamble et al., 2018).

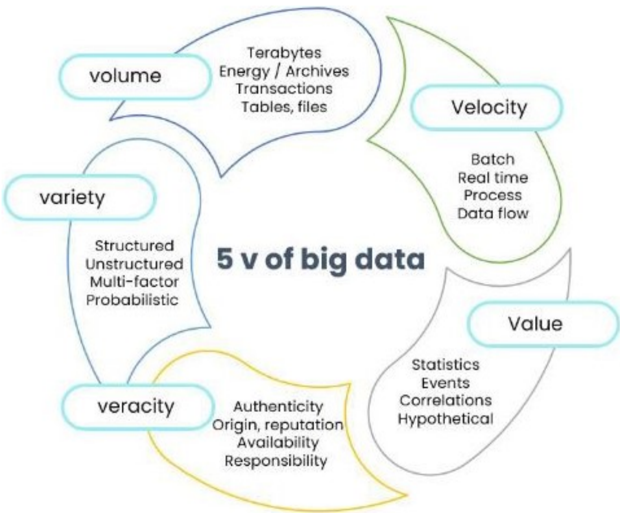


Figure 3.6: The 5V's of Big Data, reprinted from Achouch et al. (2022)

3.4.2. Cyber-Physical Systems

According to J. Lee et al. (2015), the technological developments of the past decades have made it easier and possible to use and effectively implement sensors and data acquisition systems. This resulted in the development of big data. The authors use this pillar of Industry 4.0 to develop the CPS further. This resulted in the author's proposal of a 5C architecture, figure 3.7 for applying CPS in a manufacturing application.

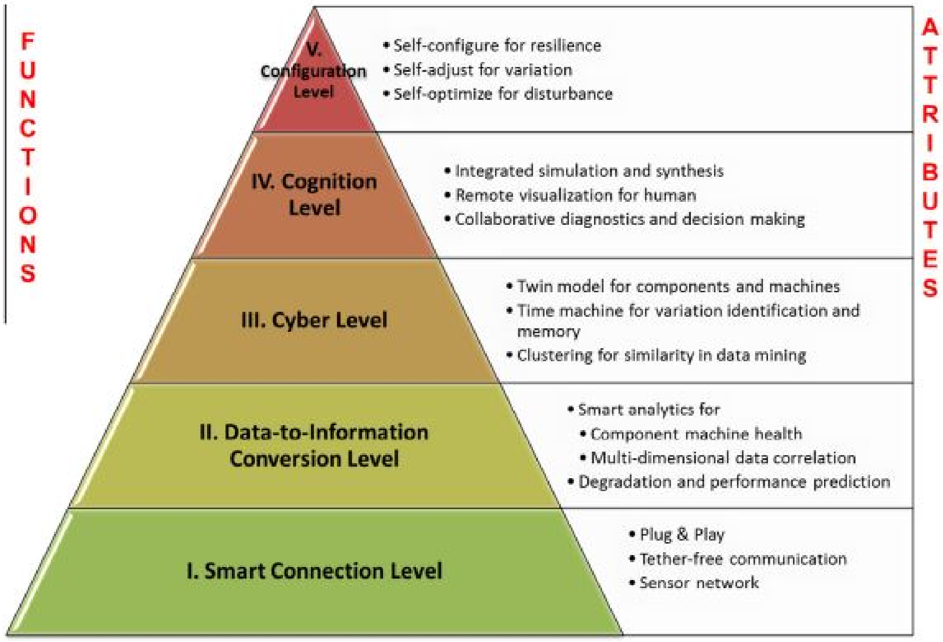


Figure 3.7: 5C architecture for the implementation of CPS by J. Lee et al. (2015)

The first level of the 5C architecture is the connection level. This level ensures useful data collection from the system and its components. This data can be collected by sensors, controllers, or enterprise manufacturing systems. The authors consider two essential factors to consider at the first level. First, the data coming from the sensors do not have to be in the same data type. Therefore, transferring all the data to one central server point could get complex. Second, considering the system's complexity, selecting suitable sensors for describing the system later on in the digital world must be considered. After collecting the data in the previous step, the next step is to translate this data into meaningful information. The methodology for converting the data to, for instance, the RUL, can differ for each system component. Frequently used is a PHM application. This level brings self-awareness to the system. The cyber level is the central hub where all information is collected from the different connected agents at the conversion level. At the cognition level, the CPS will ensure that the whole monitored system is known. The gathered knowledge can then be presented and visualized to help in decision-making tasks. The final level is the feedback from the cyber environment to the physical reality. Here, the system can be made self-configured and self-adaptive.

With the introduction of Industry 4.0 and the defined pillars, research regarding this subject has gained interest. With this growth in research, many interpretations of the terms related to Industry 4.0 have arrived. As seen with PdM and using the terms CBM, PHM and RUL in different situations, the same has happened with IoT and CPS (Lesch et al., 2023). However, what is meant when using IoT and CPS? This question was investigated by Lesch et al. (2023). Based on their research, the authors found six important terms that capture the concepts' essence, as seen in figure 3.8. Based on these terms, Lesch et al. (2023) came up with the following redefined definitions of IoT and CPS and will be used as the definition in this research:

"The Internet of Things (IoT) consists of physical entities (things) that were not necessarily intended for communication with each other and with the environment. In IoT, these things are able to identify themselves, communicate, and interact via a network, based on Internet technologies. They can act depending on external triggers or local logic."

"CPS are systems consisting of tightly integrated physical and cyber components interconnected through one or more networks. The cyber components consist of computing and communication facilities (local or remote, e.g., embedded systems or cloud services) used for monitoring, automating and controlling physical systems and processes. CPS are normally based on complex feedback and control loops, where the physical components affect the cyber components and vice versa"

Term	IoT	CPS
Communication via a network	Yes	Yes
Integration of virtual and physical world	No	Yes
Computation/process	No	Yes
Control	No	Yes
Identification/interaction	Yes	No
Environment	Yes	No

Figure 3.8: Important terms for defining IoT and CPS, reprinted from Lesch et al. (2023)

3.5. Multi-component systems

With the industry increasing continuously with the introduction of Industry 1.0-4.0, the equipment and systems used are becoming more innovative but also more complex. With these systems containing multiple components, finding a maintenance policy that reflects the actual state of the system becomes challenging. A multi-component approach must thus include all the relationships between the components to result in an accurate representation of the system. Here the multi-component approach introduces the dependencies between different components of the system (Van Horenbeek & Pintelon, 2013). Secondly, because of the multi-component system, visualizing and reasoning can become hard to understand. Therefore, it is key to result in a solution that is easy to explain and understand for the people using it (Gashi & Thalmann, 2020).

3.5.1. Component dependencies

Due to the technological advances in Industry 3.0 and 4.0, multi-component systems are becoming increasingly complex. With the complexity of multi-component systems, not considering the dependencies between the components would lead to an inaccurate description of the system and result in inefficient maintenance strategies (Van Horenbeek & Pintelon, 2013). The dependence between different components is defined in the research by Nicolai and Dekker (2008). The authors defined the interdependence of the components into four classes: economic, stochastic, structural dependence and resource dependence. Economic dependence implies that grouping the components either saves cost, positive economic dependence, when the components are jointly maintained instead of individually. Or increase the maintenance cost, negative economic dependence, when maintaining in groups is more expensive than maintaining each component separately. Stochastic dependence means that the state of component 1 has an influence on the state of component 2. This state can be defined as a condition measure, for example, deterioration or failure rate. Structural dependence between components implies that if component 1 needs to be replaced, component 2 also be disassembled to replace component 1. This can mean but is not limited to the components being part of each other. When doing maintenance activities, the availability of resources is essential but often neglected in PdM research to simplify the situation. Resource dependence needs to be taken into account when analyzing multi-component systems to ensure that the model is actually doable. By knowing what resources are needed and if always new parts are needed, or revision can take place, sustainability is also taken into account (Gashi & Thalmann, 2020).

3.5.2. Uncertainty

In the previous section, the dependency between components was explained to be considered in predicting the system's health. The same as the dependency, uncertainty needs to be considered when developing the system's prediction model (Atamuradov et al., 2020). Atamuradov et al. (2020) state that the following uncertainties must be quantified for health prognostics. Uncertainty in system parameters: this uncertainty comes from the physical system itself, where the parameters could be influenced by environmental or operational conditions. The uncertainty in translating the system condition to a mathematical model: here the mathematical model to represent the system's behavior could differ due to assumptions made for the system. The third uncertainty is the uncertainty in the degradation model used. When predicting the system's health based on the degradation, a degradation model is used based on a life test to compare the current situation with the literature. Here the obtained degradation trend could differ from the literature. Also, the uncertainty in the prediction itself needs to be considered. Although enough data could make the system accurate enough to predict, uncertainty is included. The last uncertainty to consider in the system's health is the failure thresholds used, which can differ over time and operating conditions.

3.5.3. Bayesian networks

To account for the dependencies and uncertainty of the different components, Bayesian Networks (BN) can be used to predict the system's health. BN uses conditional probability and Bayes' Theorem to represent the uncertainties and relations of the components, as seen in equation 3.1 (Gomes & Wolf, 2020).

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)} \tag{3.1}$$

where
H is the hypothesis or event whose probability was determined.
E is the evidence or the new data that can affect the hypothesis.
$P(H)$ is the prior probability or the probability of the hypothesis before the new data was available.
$P(E)$ is the marginal likelihood and probability of the event occurring.
$P(E|H)$ is the probability that event E occurs, given that event H has already occurred. It is also called the likelihood.
$P(H|E)$ is the posterior probability and determines the probability of even H when even E has occurred. Hence, event E is the update required.

The dependencies between the components are displayed with an Acyclic Directed Graph (GDA), while the knowledge of the components is displayed in conditional probability tables for discrete variables or probability density functions for continuous variables (Gomes & Wolf, 2020). An example of a BN is visualized in figure 3.9. Within the network, the investigated variables are represented as nodes. The nodes in figure 3.9 are A, B and C. Here a variable can either be discrete or continuous. The link between the nodes represents the dependence relation between the variables. In the BN, node (B) is the parent of node (A) if there is a link from node B to node A. A is then the child of node B (and in figure 3.9 also from node C). A node without parents is called a root node, and a node without a child is called a leaf node. The conditional independence of the BN results in that the chain rule can be used to calculate the joint probability of a node. This leads to the following formula:

$$P(X_1, X_2, ..., X_n) = \Pi_{i=1}^{n} P(X_i|pa(X_i)) \tag{3.2}$$

where $pa(X_i)$ is the set of parent nodes of X_1 (D. Lee & Pan, 2017)

To include time in a BN, a DBN can be used, figure 3.10. DBNs can model the temporal dependencies between the variables. Here the model starts with a BN, and in the next time step, the network is recreated to obtain a dynamic model. The relationship between variables in the same time frame is called intra-slices, while the relationship of variables between different timeframes is called inter-slices. The temporal dependencies are drawn as a dashed link between the variables in the other time slices of the DBN, as presented in figure 3.10 (Zhao et al., 2020).

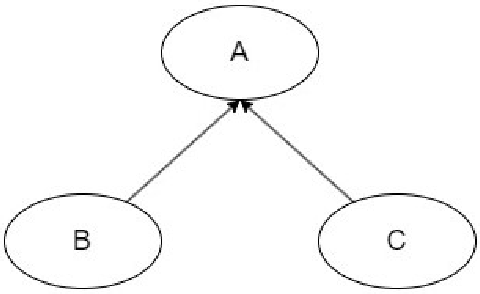


Figure 3.9: An example of a Bayesian Network

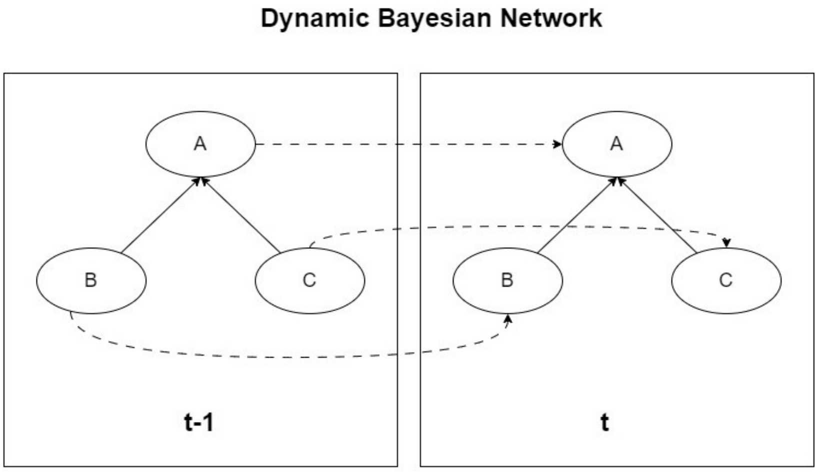


Figure 3.10: An example of a Dynamic Bayesian Network

3.5.4. Grouping maintenance activities

Economic dependences between components can be included in the BN by enabling maintenance grouping. In the paper of Hu et al. (2012), the authors propose an opportunistic PdM strategy at the component and system levels based on economic dependencies. With these opportunistic strategies, the optimal time to do maintenance is found based on the expected cost rate as shown in figure 3.11.

For local optimization, three types of cost rates need to be considered. First, the expected repair cost rate represents the cost for all related to the repairment of the component, for example, spare parts or manpower cost. The expected repair cost rate can be divided into two parts: corrective costs and proactive costs. The corrective cost is the cost if the component fails before it is expected to fail and thus immediately needs to be repaired. The proactive cost is the repair cost if the repair is scheduled. Usually, the repair cost is more expensive if corrective costs must be considered due to not preparing and expecting the repair job. The repair cost rate is determined as the sum of the expected corrective and proactive repair cost divided by the time spent since the last maintenance moment Δt, where $\Delta t = t - t_0$:

$$RC_{r\_i}(t) = \frac{RC_i^c F_i(t) + RC_i^p (1 - F_i(t))}{\Delta t} \tag{3.3}$$

where RC_i^c is repair cost for corrective repair, RC_i^p the cost for proactive repair, F_i the failure probability distribution of component i. It represents the cumulative distribution function of the random variable “time to failure”. As the cost varies randomly according to the economic and labor resources, a distribution function for the repair cost for corrective and proactive repair is used.

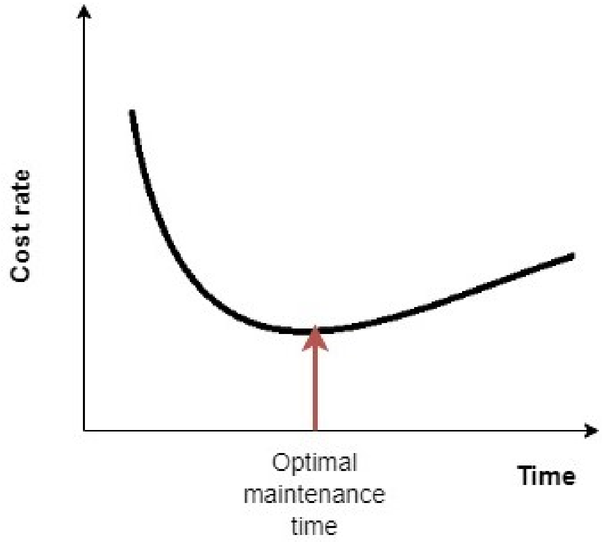


Figure 3.11: Example of the expected cost rate

The cost of setting up the repair is the second part of the expected total cost rate. The set-up cost can, for instance, contain the mobilization cost for the repair crew, tools, and machine disassembling. The expected set-up cost rate is calculated as follows:

$$SC_{r\_i}(t) = \frac{SC}{\Delta t} \tag{3.4}$$

Here, SC stands for the set-up cost of the system and is also assumed to change randomly with a normal distribution.

The third rate is based on the cost of production losses, the cost of not being able to run the system. The production losses are divided into two parts: the losses for corrective repair activities and those due to proactive repair activities. Here, also, the assumption is made that the losses for corrective repair are greater than for proactive repair.

$$PL_{r\_i}(t) = \frac{DT_i^c L^c F_i(t) + DT_i^p L^p (1 - F_i(t))}{\Delta t} \tag{3.5}$$

Where DT_i^c is the out-of-order time due to corrective repair for component i, and DT_i^p is the out-of-order time due to proactive repair for component i. L^c and L^p are the losses per unit of time. Also, here, the variables are assumed to change randomly with a specific distribution.

Combining the three types of costs, the expected total cost rate for component i, will be:

$$C_{r\_i}(t) = RC_{r\_i}(t) + SC_{r\_i}(t) + PL_{r\_i}(t) \tag{3.6}$$

i.e.

$$C_{r\_i}(t) = \frac{RC_i^c F_i(t) + RC_i^p (1 - F_i(t))}{\Delta t} + \frac{SC}{\Delta t} + \frac{DT_i^c L^c F_i(t) + DT_i^p L^p (1 - F_i(t))}{\Delta t} \tag{3.7}$$

With this formula, the local optimal PdM time can be calculated by minimizing this objective function for each component.

The global opportunistic PdM strategy by Hu et al. (2012) tries to group maintenance activities to save money and time by doing maintenance activities jointly instead of separately. In this way, costs due to the unavailability of the system and setup costs can be reduced. By using maintenance group G, consisting of k components, the three cost rate functions of the local optimization strategy are adjusted to:

$$RC_{r\_G}(t) = \frac{\sum_{i=1}^{k}(RC_i^c F_i(t) + RC_i^p(1 - F_i(t)))}{\Delta t} \tag{3.8}$$

$$SC_{r\_G}(t) = \frac{SC}{\Delta t} \tag{3.9}$$

$$PL_{r\_G}(t) = \frac{(\sum_{i=1}^{k}(\alpha DT_i^c L^c F_i(t) + \beta DT_i^p L^p(1 - F_i(t)))}{\Delta t} \quad \text{with} \quad 0 \leq \alpha \leq 1 \quad \text{and} \quad 0 \leq \beta \leq 1 \tag{3.10}$$

α and β are here introduced as reduction coefficients due to the grouping of maintenance activities, which results in fewer outage durations. Here, the authors assume that outages for group durations are usually less than the summation of the outage durations of the components. Here α and β are determined by historical data or expert knowledge. The total expected cost per unit of time for the group G is calculated as:

$$C_{r\_G}(t) = RC_{r\_G}(t) + SC_{r\_G}(t) + PL_{r\_G}(t) \tag{3.11}$$

With these formulas, the optimal PdM strategy and the corresponding groups are determined as follows: First, the local opportunistic PdM time t_i^{op} is calculated for each component. This optimal time sequence, $t^{op} = \{t_1^{op}, t_2^{op}, ...t_{NC}^{op}\}$ with NC as the number of components, is then rearranged in ascending order: $t_m^{op} = \{t_{m1}^{op}, t_{m2}^{op}, ...t_{mNC}^{op}\}$ with $t_{m1}^{op} = min\{t_1^{op}, t_2^{op}, ...t_{NC}^{op}\}$ and $t_{mNC}^{op} = max\{t_1^{op}, t_2^{op}, ...t_{NC}^{op}\}$. Then, the first component with t_{m1}^{op} is placed in the maintenance group G, and the minimal maintenance cost rate is calculated. Next, the second component j, with t_{m2}^{op}, in the sequence is placed in group G, and the function is again minimized. With the components outside the group considered as repaired independently, the expected total minimal maintenance cost rate is calculated as:

$$C_{systemrate_{op}} = C_{r\_G} + \sum_{i=j+1}^{NC} C_{r_{mi}}^{op} \tag{3.12}$$

If the expected total minimal maintenance cost rate is lower than the previously calculated expected total cost rate, the maintenance group can be expended; otherwise, the optimal group is found, and the corresponding maintenance time sequence can be determined as $t_{sys} = \{t_G^{op}, t_{m(j+1)}, ...t_{mNC}^{op}\}$.

3.5.5. Imperfect maintenance

Although the maintenance mechanics are trying to do their maintenance work as best as possible, it can not be assumed that after repairments, the probability of failure is zero percent. Therefore, Van Horenbeek and Pintelon (2013) introduce imperfect maintenance in their model. Here, each maintenance activity reduces the degradation level of a component by a factor (1-B), with 0 ≤B ≤1. With this factor B, named the improvement factor, a more realistic maintenance approach can be used.

3.6. Conclusion

In this chapter, a literature survey about maintenance strategies has been executed, focusing on PdM and its relation to multi-component systems and Industry 4.0. As the complexity increases with more components, it is critical to ensure an easy-to-understand solution. Therefore, combinations of the three single-model approaches are recommended when developing a PdM strategy. Here, the advantages of each model can be used to reduce the system's complexity. Second, the objective of a PdM strategy of a multi-component system can be more than one. It could be that the objectives are conflicting within the system, and a trade-off must be found. With the different components within the system, the relationships between the components must be included to represent the system correctly. These relationships and predictions of the health uncertainties must be included to know how accurate the multi-component system model will be. Here, grouping maintenance activities and incorporating imperfect maintenance will make the model more realistic. PdM can be enhanced with the upcoming Industry 4.0 and the nine pillars. Especially in data-driven methods, big data and the use of sensors enable the monitoring of a system at all points. However, this translates into enormous data storage and unnecessary data collection. With the complexity and size of multi-component systems, a well-defined architecture is needed to ensure that the correct data is collected. From the literature, it became clear that the development of a PdM strategy for multi-component systems is still in the beginning phase compared to single-component models. As the complexity increases with more components, the literature focuses on parts of the system while making assumptions about the other parts to reduce the complexity. This results in the use of mostly theoretical models, and the connection to the real world is (partly) absent. This connection can be formed by combining the theoretical models with the pillars of Industry 4.0.

Based on the literature survey, the gap in the literature is defined on the following points. First, no papers regarding PdM for PBBs could be found. As the literature on PdM is still in the beginning phase for multi-component systems, the number of papers regarding implementation in practice is also limited. This results in a literature gap in developing a PdM strategy for multi-component systems in practice. Also, the evaluation of this strategy based on real-world data is limited. Most literature uses preset datasets where the data is already classified in the faulty and normal behavior of the system. However, as was seen for the PBB, data can be collected by using sensors, but knowing what it will represent, faulty or normal behavior, can be difficult. Finally, with the technological developments within Industry 4.0, it was concluded from the literature that it could enhance the PdM strategies in a data-driven way. Here, the gap of combining this with a multi-component system can be overcome with this research. In the following chapter, a CPS architecture for a multi-component system is introduced to overcome the previously mentioned points.

4 CPS Architecture for a Multi-Component System

In this chapter, the proposed architecture for a PdM strategy of a multi-component system is presented. This chapter answers the sub-question: "How can the prediction model be developed?". In the first section, the choices made for the proposed architecture are explained. In the following sections, the architecture is explained step by step.

4.1. Methodology selection

The proposed architecture uses the CPS architecture as presented in J. Lee et al. (2015). The choice for a CPS architecture and not an IoT environment was based on the fact that, in the end, the system can be autonomous by integrating computing, monitoring and control of the physical sub-systems. By enabling the system to affect the health status of the sub-systems in the cyber part and to be able to do maintenance activities for the physical part. The system enables itself to control its health. While only using an IoT environment, the feedback loop to the physical part is not made, and the potential already incorporating the step to autonomous assets still needs to be made in the future. With the CPS architecture, a PHM system is created to manage the system's health. By choosing a PHM system instead of only applying condition-based maintenance, the system is monitored and proactively managed to acquire a reliable system. Within this PHM system, the choice has been made to use a low-level model and a higher-level model. The lower-level model determines the health status of the sub-systems. The choice for this lower-level model has been made because the PHM system continuously tries to keep the health status as high as possible. Therefore, it cannot rely only on the determined system health; insight must also be created into the sub-systems' health. This is then combined with that from the literature; it was found that to have integration in practice, it must be explainable and easy to understand. Having insight into the sub-system health, the complex system will be subdivided into multiple easier-to-understand sub-systems. To make the multi-component system also less complex, section 3.2.4 showed that using a multi-model approach can reduce the complexity by combining multiple models. As concluded in chapter 2, degradation is not the primary cause of failure, and multiple failure mechanisms are in play. This results in multiple models that need to be used to determine the health status of the sub-systems. Therefore, a multi-model approach is proposed. With this approach, the complexity of the system and sub-systems are kept in mind before selecting the methods for determining the health status. The higher-level model builds upon the lower-level model and determines the health status of the system. A DBN is used to incorporate the sub-system dependencies. By developing a DBN, insight is given into the system's complexity, and an estimation of the system's health status can be given. Using the DBN, the failure mechanisms, which cannot be linked to a sub-system directly, can be included in the network. By predicting the health status over time, unknown influences can be modeled as interslice links between the system's sub-systems. As cost minimization is not the research goal, the PdM strategy object, as discussed in section 3.3, will not have a cost optimization. In this research, the optimization criteria will be reliability and availability optimization. This means that after the health status is determined, decision-making occurs in finding an optimum between minimizing the total out-of-order time of the system but limiting the maintenance moments and thus maximizing the aircraft stand availability while maintaining a reliability threshold for the PBB.

4.2. Connection layer

The first layer of the chosen CPS architecture is the connection layer. This layer collects data from different sources to connect the physical object with the virtual world. When developing a PdM strategy, what needs to be monitored to make a prediction, either on the sub-system level or on the system level, must be known. When developing a prediction model for a single component, one can already have a clear vision of what to monitor and what the possible failure mechanism is. However, as seen in section 3.5, a multi-component system makes the prediction more complex. This is combined with the fact that the investigated system is more likely to fail due to different reasons than degradation or wear, as seen in section 2.4. Although the technological developments of Industry 4.0 have given an enormous amount of data available, collecting all data that could be potentially of interest is a difficult job. Therefore, the first step in the connection layer of the proposed CPS architecture consists of using the information from the analyzed maintenance tickets and experts' knowledge to determine the root causes of failure. The maintenance logs within the maintenance tickets are also interesting in the proposed architecture. The maintenance logs are a free text box within the maintenance ticket where the mechanic can write down what the root cause of the failure was and how the problem was solved. The maintenance logs can be analyzed in multiple ways; the most straightforward way is analyzing all maintenance logs by hand. However, analyzing by hand will also be time-consuming and impractical in the context of Big Data. To be more time efficient, a Natural Language Processing (NLP) algorithm can be used to analyze the maintenance logs (Akhbardeh et al., 2020; Sharp et al., 2017). NLP is a part of artificial intelligence where computers derive understanding from human language by analyzing it cleverly and usefully (Usuga-Cadavid et al., 2022). By using NLP, the first step is pre-processing the maintenance log data. The maintenance logs are normalized by checking the text on stopwords and punctuation, including lowercasing and removing special characters. Normalization needs to be done because the logs are analyzed to obtain more insight into the reasons behind the bridge's failure. It is therefore not wanted to see that articles are the most frequently occurring words in the logs. This results in the use of stopword removal. The same reason can be applied to the lower casing of the logs and the removal of punctuation and special characters. Then, the logs are tokenized, and with the Bag of Words principle, the most frequently occurring words or sequences can be extracted. As stated, state-of-the-art technology that has seen a lift by Industry 4.0 is Artificial intelligence. A new development in the field of AI is OpenAI. The OpenAI software, which can be used for the maintenance logs analysis, uses a large language model (LLM). Prompt engineering techniques are used to use this LLM and increase its functionality. A prompt is the instruction given to the LLM in which the model finds the answer for (White et al., 2023). Prompt engineering is the methodology of finalizing the LLM prompts to give the best (precise, coherent or accurate) output. With prompt engineering, the prompts are fine-tuned to realize the output results, which are wanted (Lo, 2023).

A prompt can be constructed based on four elements: instruction, context, input data and output indicator. The instruction part contains the specific goal the model needs to achieve. Context can be provided to guide the model in the right direction. The input data is the input or question where a response is necessary. Also, an output indicator can be given to tell the model how the output needs to be displayed (DAIR.AI, 2023). Multiple techniques can be used for prompt engineering; an in-depth overview of these techniques can be found in DAIR.AI (2023):

- Zero-shot prompting
- Few-shot prompting
- Chain-of-Thought Prompting
- Self-Consistency
- Generate Knowledge Prompting
- Automatic Prompt Engineer

- Active-Prompt
- Directional Stimulus Prompting
- ReAct
- Multimodal Chain-of-Thought
- Graph Prompting

For the experts' knowledge, the maintenance mechanics can be consulted. They are constantly involved in the maintenance activities of the system. They know from their daily operations what subsystems and related failure mechanisms are of interest when predicting the system's health status. Based on the maintenance logs and expert knowledge, data can be extracted from the available data sources.

The second step in the connection layer is the actual data extraction from the different data sources. Potential data sources are, for example, a PLC, historical data sources, sensors, and information regarding operational conditions from experts or maintenance logs. Cloud storage could be used to store the data. In figure 4.1, the first layer of the proposed architecture is schematically displayed.

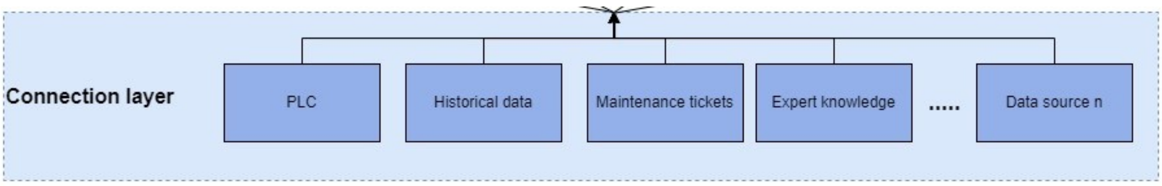


Figure 4.1: The connection layer of the proposed architecture

4.3. Conversion layer

The data collected in the first layer are the raw signals directly coming from the system. These raw signals are collected at a high frequency and are large in amount. Feature extraction and dimension reduction ensure that the collected data becomes valuable information. First, the large amount of collected data must be divided into smaller parts for signal analysis. For the analysis of the signals, three domains for analyzation can be classified:

- Time domain
- Frequency domain
- Time-frequency domain

As the names already indicate, analyzation in the time and frequency domain looks at how the signal variates over time or frequency. The time-frequency domain is used for non-stationary signals, where converted to the frequency domain, no information can be extracted if the assumption is made that the frequency sub-systems do not vary over time (Calabrese et al., 2021; Gawde et al., 2023). As each component under investigation can have a different reason for failure, the feature extraction and dimensionally reduction techniques can differ from component to component. Therefore, only the used feature extraction and dimensionality reduction methods will be explained in the next chapter; an overview of different techniques can be found in Calabrese et al. (2021) and Gawde et al. (2023).

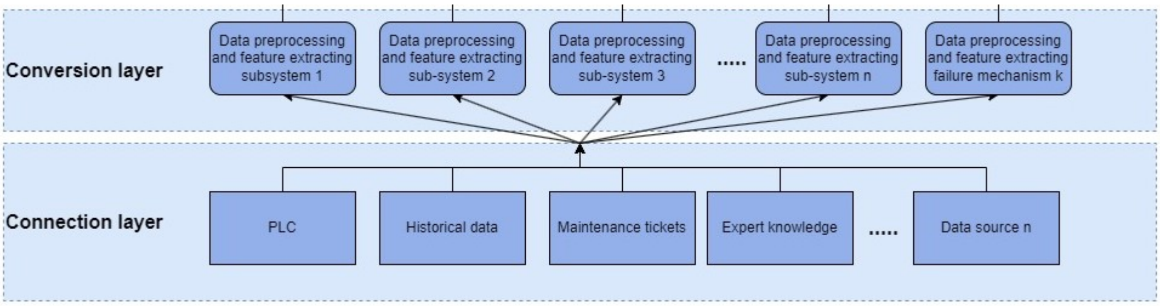


Figure 4.2: The conversion layer of the proposed architecture

4.4. Cyber layer

From the analysis in the first layer, it became clear what needed to be monitored. These indicators are then split into two categories. The first group is indicators of a component's health. These components are part of a sub-system. The second group is indicators that cannot be related to a sub-system and, therefore, are classified in the model as general failure mechanisms and thus directly related to the system's health. Each sub-system's health status prediction individually takes place in the cyber layer. In this stage, no dependencies or correlations between different sub-systems will influence the investigated sub-system's predicted health status.

For the sub-systems, there can be many factors as to why the component(s) within fails. Therefore, the sub-systems were analyzed in the first layer to state which sub-systems failed during the system's operation. Within this analysis, the second step was finding what components caused the sub-system to fail. Because the system can also fail due to reasons outside only degradation of the sub-systems based on life cycles, as seen in section 2.4, the failure mechanisms, either related to the sub-system or directly to the system, must be monitored too. As the amount of data and models could increase with the increment of the failure models, a multi-model approach, as presented in figure 3.2.4, is suggested. However, this depends on the specific sub-system and failure modes. The first part of this multi-model approach is based on the acquired data collected based on expert knowledge and the maintenance ticket analysis. From this analysis, the possibility occurs where not one but multiple components of the sub-system are responsible for the failure of the sub-system. The prediction model approaches, which could be used in this part, are explained in section 3.2.

The second model within the multi-model approach for the sub-system's health assessment is a data-driven model to assess the degradation of the sub-system based on expert knowledge or a degradation model from the literature. As discussed in section 2.4 for the PBB but also applicable for other systems in the industry, it does not mean that degradation is not one of the causes of failure if these faults are not occurring. As suggested for the PBB, it could be that the preventive maintenance activities are so well executed that failures due to system aging are not present. However, one of the purposes of developing a PdM strategy is to, in the end, no longer have to execute the time-based preventive maintenance strategies. Therefore, a model which assesses the sub-system health based on degradation is needed.

Within the cyber layer, the prediction model for assessing the sub-system's health will be trained by historical data to understand the behavior of the bridge. This results in the sub-system agents having an offline and online phase within the cyber layer. Figure 4.3 depicts the framework for these two phases.

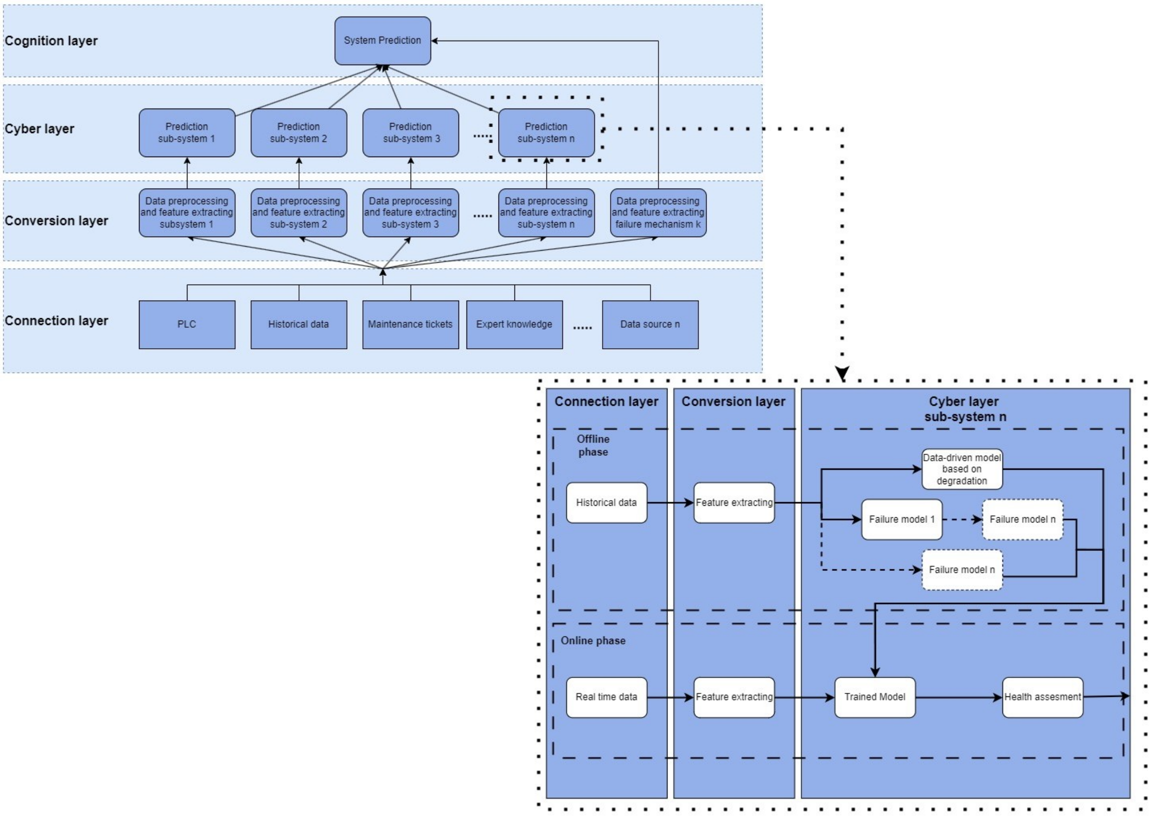


Figure 4.3: The cyber layer of the proposed architecture

4.5. Cognition layer

At the cognition level, the system as a whole will be evaluated to determine the health status. A BN, introduced in chapter 3, will be used. First, the nodes need to be identified. The variables represented by the nodes in the DBN can be categorized into four groups: component nodes, sub-system nodes, failure mechanism nodes and system nodes. In this CPS architecture, the health status of the sub-system based on the failure mechanism of the different underlying components is already determined in the cyber layer. However, it could be possible that for some sub-systems, the health status could not be determined due to hidden failure modes. Therefore, the sub-system is represented in the BN by individual nodes. The failure mechanism nodes contain variables not linked to a specific sub-system based on the maintenance ticket analysis or expert knowledge. Here, this node directly influences the system's health status. Second, the dependencies and correlations of the different sub-systems must be considered. As explained in section 3.5, the dependencies between the different sub-systems need to be considered to have an accurate description of the system. The relationships are added by developing the network structure of the BN and adding the links between nodes. Now, the BN gives an easy-to-understand overview of the complex system. At last, the CPTs of the variables need to be determined. Here, the health status is already defined in the previous layer, and the other nodes can be determined by historical data or expert knowledge to be, in the end, updated while running the model.

Next, the DBN will be created. A general indication of the timeslices is visualized in figure 4.4. The approach for the DBN is based on the model of Gomes and Wolf (2020). This means that the DBN has three timeframes. At the current time frame, evidence is presented to the system based on determining the various health status in the cyber layer. This allows the DBN to estimate the current state of the sub-systems and the system. This means that the current time frame is responsible for the diagnosis part. It also uses the previous timeframes with the evidence given to the system at that timestep as temporal evidence. The next timeslices are used for prognostic purposes. In this DBN model, temporal links are placed between each node in the same place at each timeslice to represent the degradation of the components, sub-systems and system. Here, the estimation can be more accurate by providing evidence to the system.

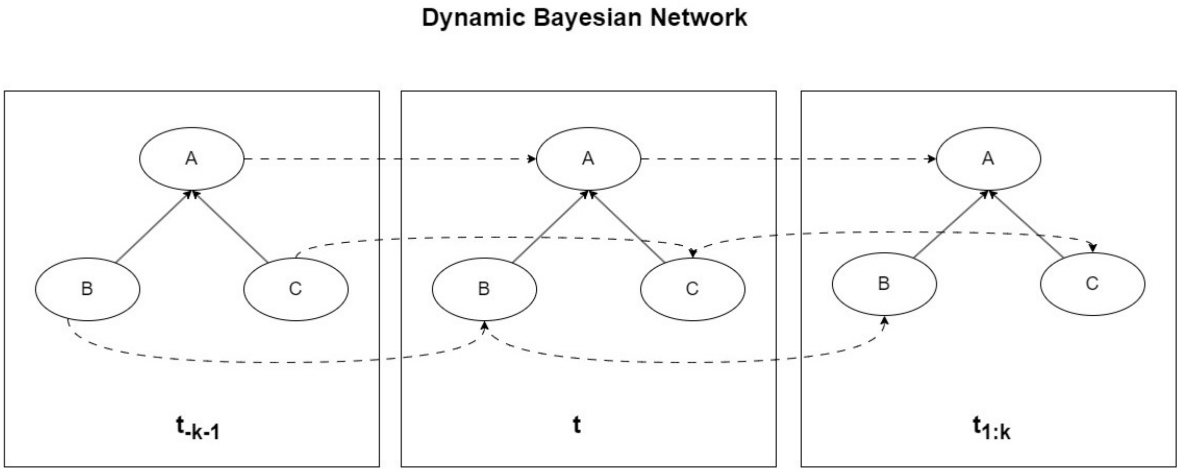


Figure 4.4: Proposed three timeframes DBN model, based on Gomes and Wolf (2020)

4.6. Configuration layer

At the top level of the proposed CPS architecture, decisions are made for when to do maintenance. Decision-making for the multi-component system can be done in various ways. The first way is decision-making by an expert. Here, the predictions made on the sub-system and system levels are assessed, and based on the displayed levels, the expert decides when to plan the maintenance activities. This real-time decision-making can also be translated into an autonomous system based on expert knowledge. In this case, the expert will give weight factors for the different sub-systems or cases so the decision model becomes a knowledge-based model.

Also, the decision-making can be done autonomously based on the DBN network. To account for the economic dependences between nodes, opportunistic PdM, the grouping of maintenance activities can be done, as explained in section 3.5.4. However, in this model, the assumption each repair restores the sub-system to the ”as good as new” state is not made. The imperfect maintenance assumption is already incorporated in the previous layer. As at each timestep the model updates the health statuses based on evidence, it is directly seen from the output of the sensor if the repairment in the previous timestep is executed well enough to overcome the failure. While the grouping of maintenance activities can be done for economic reasons, the availability and reliability of the system are of interest in this research. With the availability, the availability of the aircraft stand is meant. This availability of the aircraft stand is of high value to Schiphol when scheduling flights. However, by only scheduling flights, the reliability of the PBB will drop, and sudden failure can occur, thus causing a delay in the turnaround process. Therefore, the opportunistic PdM strategy is reformed: The reliability of an element within the DBN is set by establishing a threshold. The higher the threshold that will be used, the earlier the maintenance activity must occur. To reform the cost model to prioritize the availability of the aircraft stand and the reliability of the PBB, the cost objective function of Hu et al. (2012) is investigated per cost part and transformed into an availability measure. As in the paper of Hu et al. (2012), first, an opportunistic PdM strategy on the local scale will be created and then one at the global level. The local opportunistic PdM strategy looks at the optimal time to maintain each sub-system individually. The global opportunistic PdM strategy looks at the system as a whole to find the optimal time to do maintenance.

4.6.1. Local opportunistic decision-making

The expected repair cost rate will be replaced by the expected repair rate (RR), formulated in equation 4.1. The expected repair rate is related to the time it costs to repair the specific sub-system during maintenance divided by the bridge uses that have passed since the last maintenance moment. The expected repair rate can be distinguished into two elements: the corrective repair time (CRT) and the proactive repair time (PRT). The corrective repair time is related to the repair time endured when sudden failure happens. The proactive repair time is associated with the repair time of the sub-system during planned maintenance. Here, it is assumed that the proactive repair time is less than the corrective repair time because when a sudden failure of the sub-system happens, an investigation needs to be done into what caused the failure. The related materials need to be gathered where planned maintenance activities can be prepared beforehand. Multiplying the failure probability with the corrective repair time reflects the expected maintenance duration when a failure occurs. The higher the probability, the more influential the impact of the corrective repair on the total out-of-order time. Multiplying the proactive repair time with the complement of the failure probability, the influential impact on maintenance executed when still working is considered.

$$RR_i(t) = \frac{CRT_i * F_i(t) + PRT_i(t) * (1 - F_i(t))}{\Delta t} \tag{4.1}$$

where

i	is the index for the sub-systems.
CRT_i	is the corrective repair time for sub-system i.
PRT_i	is the proactive repair time for sub-system i.
F_i	represents the failure probability distribution of sub-system i.
Δt	is the time spent since the last maintenance moment.

The expected setup cost rate is replaced by the expected arrival rate (AR). The expected arrival rate is related to the time it costs for the maintenance mechanism to arrive at the PBB, specified as AT, divided by the time spent since the last maintenance moment. The arrival time needs to be included as it is important to have the mechanic as fast as possible at the site to prevent delay due to slow response time. By including the arrival time within the expected arrival rate, the influence of the mechanics response can be captured within the model.

$$AR_i(t) = \frac{AT_i}{\Delta t} \tag{4.2}$$

where

i	is the index for the sub-systems.
AT_i	is the arrival time for sub-system i.
Δt	is the time spent since the last maintenance moment.

The expected loss of productivity is replaced with the time the system is out of order due to corrective repair or proactive repair, which is the same as the expected repair rate. Therefore, the time of productivity loss is not considered separate but included in the repair time.

Second, resource availability must be considered. For resource availability, the work schedule of the maintenance mechanics can be incorporated into the model. The work schedule must specify how many workers are available at each time step. Here, it is assumed that the capacity of the workers available will differ over time. For the repair times, the assumption is made that a repair of one or multiple sub-systems will be faster with multiple mechanics than with one. Here, a distinction is made between repairs at the site and the time related to the sub-system repair time at the workshop. In the case of proactive repair, this can happen before the actual repairments are done. The different factors for speeding up the process are alpha for the site and beta for the shop. The formula for the repair rate is therefore expanded to:

$$CRT_i(t) = \frac{CRT_i^{site}}{\alpha * m(t)} + \frac{CRT_i^{shop}}{\beta * m(t)} \tag{4.3}$$

where

i	is the index for the sub-systems.
CRT_i^{site}	is the corrective repair time at the site for sub-system i.
α	the reduction coefficient when using multiple mechanics at the site.
CRT_i^{shop}	is the corrective repair time at the shop for sub-system i.
β	the reduction coefficient when using multiple mechanics at the shop.
$m(t)$	is the amount of maintenance mechanics at timestep t

$$PRT_i(t) = \frac{PRT_i^{site}}{\alpha * m(t)} + \frac{PRT_i^{shop}}{\beta * m(t)} \tag{4.4}$$

where

i	is the index for the sub-systems.
PRT_i^{site}	is the proactive repair time at the site for sub-system i.
α	the reduction coefficient when using multiple mechanics at the site.
PRT_i^{shop}	is the proactive repair time at the shop for sub-system i.
β	the reduction coefficient when using multiple mechanics at the shop.
$m(t)$	is the amount of maintenance mechanics at timestep t

In practice, it could be that instead of taking the sub-system out of the system and then to the workshop for a repair, the sub-system is replaced at the site to reduce the out-of-order time of the system. Therefore, the time of the workshop part could be zero in this case.

Combining the expected repair rate with the expected arrival rate, the expected total out-of-order rate for sub-system i (TR_i) is represented by:

$$TR_i(t) = RR_i(t) + AR_i(t) \tag{4.5}$$

i.e.

$$TR_i(t) = \frac{(\frac{CRT_i^{site}}{\alpha * m(t)} + \frac{CRT_i^{shop}}{\beta * m(t)}) * F_i(t) + (\frac{PRT_i^{site}}{\alpha * m(t)} + \frac{PRT_i^{shop}}{\beta * m(t)}) * (1 - F_i(t))}{\Delta t} + \frac{AT_i}{\Delta t} \tag{4.6}$$

By minimizing the expected total rate it takes to do the maintenance activities, the local optimal PdM moment can be determined. With this minimization, the optimal time to do maintenance is found where the expected total repair rate for each sub-system individually is at its lowest. Minimizing the repair rate creates a trade-off between correcting the sudden failure and proactively maintaining the still-working sub-systems. Including the multiplication of the proactive repair time times the complement of the failure probability prevents the maintenance activities from being scheduled too early. If the failure probability is low, executing maintenance activities too early could increase the overall maintenance time. However, when planning maintenance activities too late, the sub-systems' failure probability will be too high to be acceptable for the airport. Therefore, a preset reliability threshold will be used. By including the expected arrival rate, the whole process from an assigned maintenance mechanic to a functioning PBB is considered in the local opportunistic model.

4.6.2. Global opportunistic decision-making

At the system level, the global opportunistic PdM strategy, as introduced in Hu et al. (2012), is applied. The local opportunistic PdM strategy looked at each sub-system individually and optimized the maintenance moments for each sub-system. The global strategy looks at the system as a whole. By using the opportunity to do maintenance not at one sub-system but at multiple sub-systems during one planned maintenance activity, the out-of-order moments of the system are reduced. This results in maintenance groups being formed. The maintenance group can consist of k sub-systems to do maintenance at one moment, as explained in section 3.5.4. By grouping maintenance activities, there is the possibility that the repair time is decreased due to the structural dependence of the sub-systems. More likely, there will be no decrease in repair time but an increase due to the sum of the individual maintenance moments. Therefore, a decision needs to be made to have multiple short periods of maintenance activities or one more extended session but an increase in availability for the rest of the time. The procedure for finding the system optimum and global PdM strategy is as follows: First, the formulas for the expected repair rate of the group, the expected arrival rate of the group and the expected total rate will be explained for group G. Thereafter, the procedure to determine the optimal group G.

First, the expected repair rate of the group (RR_{group}) is calculated. This will be done by summing up the corrective and proactive repair times of the sub-systems k in optimal group G. A reduction coefficient omega is used. With this omega, the influence of grouping sub-systems where, due to structural dependencies, the repair time could be reduced is taken into account.

$$RR_{group}(t) = \frac{\sum_{i=1}^{k}(\omega_{CRT}CRT_i(t) * F_i(t) + \omega_{PRT}PRT_i(t) * (1 - F_i(t)))}{\Delta t} \tag{4.7}$$

where

i	is the index for the sub-systems.
k	is the number of sub-systems in group G.
CRT_i	is the corrective repair time for sub-system i as presented in equation 4.3.
PRT_i	is the proactive repair time for sub-system i as presented in equation 4.4.
F_i	represents the failure probability distribution of sub-system i.
Δt	is the time spent since the last maintenance moment.
ω	is the reduction coefficient when grouping sub-systems.

Second, the expected arrival rate of the group is calculated. Here, the arrival time for doing maintenance activities for group G will be divided by the time spent since the last maintenance moment.

$$AR_{group}(t) = \frac{AT_{group}}{\Delta t} \tag{4.8}$$

where

AT_{group}	is the arrival time for the group.
Δt	is the time spent since the last maintenance moment.

The same as in the local strategy, the expected total rate of the group is the expected repair rate of the group together with the expected arrival rate of the group. By minimizing the expected total rate, the optimal time to do maintenance for the group can be found.

$$TR_{group}(t) = RR_{group}(t) + AR_{group}(t) \tag{4.9}$$

where

RR_{group} is the repair time rate for the group
AR_{group} is the arrival time rate for group G.

From the local opportunistic strategy, the optimal time to do maintenance for each sub-system is determined and defined as t_i^{op}. The optimal time sequence to do maintenance is $t^{op} = \{t_1^{op}, t_2^{op}, ...t_{NC}^{op}\}$ with NC as the number of sub-systems, is then rearranged in ascending order: $t_m^{op} = \{t_{m1}^{op}, t_{m2}^{op}, ...t_{mNC}^{op}\}$ with $t_{m1}^{op} = min\{t_1^{op}, t_2^{op}, ...t_{NC}^{op}\}$ and $t_{mNC}^{op} = max\{t_1^{op}, t_2^{op}, ...t_{NC}^{op}\}$. To create the group, the first sub-system with t_{m1}^{op} is placed in the maintenance group G, and the expected minimal total maintenance rate of the group is calculated with equation 4.9. Next, the second sub-system j, with t_{m2}^{op}, in the sequence is placed in group G, and the function is again minimized. With the sub-systems outside the group considered as repaired independently, the expected total minimal maintenance rate (TR_{sys}) of the system is calculated with equation 4.10. If the expected total maintenance rate of the system based on the new maintenance group consisting of the first and second sub-systems is less than the expected total maintenance rate based on adding the total rates of the individual sub-systems, the next sub-system with t_{m3}^{op} is added to the group and a new total rate is calculated. This procedure will be repeated until the newly formed group has a higher total maintenance repair rate of the system than the current determined group. Then, the current group will be the optimal group for combining maintenance activities, with the other sub-systems being repaired independently.

$$TR_{sys} = TR_{group}^{opt} + \sum_{i=k+1}^{NC} TR_i^{opt} \tag{4.10}$$

where

G is the index for the group.
k is the number of sub-systems in group G.
NC are the non-group sub-systems.
TR_{group}^{opt} is the total repair rate for group G at the optimal time to do maintenance.
TR_i^{opt} is the total repair rate for sub-system i at the optimal time to do maintenance.

4.7. Conclusion

In this chapter, a CPS architecture is proposed for developing a predictive maintenance strategy for a multi-component system. With the CPS architecture, a lower-level model and a higher-level model is used. The lower-level model determines the health status of the sub-systems. By first addressing the root causes of the system's failure, adequate data collection can be done, enabling continuous health monitoring of the sub-systems. The higher-level model builds upon the lower-level model and determines the health status of the system. A DBN is used to incorporate the sub-system dependencies in the cognition layer. In the final layer of the CPS architecture, decision-making takes place. This research prioritizes reliability and availability optimization over cost minimization. This means that after the health status is determined, decision-making takes place in finding an optimum between minimizing the total out-of-order time of the system but limiting the maintenance moments and thus maximizing the system's availability while maintaining a reliability threshold. In the next chapter, the proposed architecture is implemented to prove the architecture and to see how the developed predictive maintenance strategy will perform in relation to the turnaround process.

5 Simulation model

In this chapter, a simulation model will be implemented to prove the architecture from the previous chapter. A simulation model has been used to obtain results instead of an implementation at Schiphol because of the lack of data and time constraints. However, with the results, the benefits of the proposed architecture will be shown for a range of parameters. The simulation model is the process executed in the higher level model with as input the output of the lower level model. A simplified model is used for the lower-level model to provide maintenance decision-making. This chapter answers the sub-question: ”How can the developed prediction model be implemented in the maintenance strategy of the Passenger Boarding Bridge?”.

5.1. Bounderies & Assumptions

In the current analysis, the seven main sub-systems of the PBB were defined by analyzing the FMECA and the SAT protocol. The different clusters for the failures were seen by analyzing the maintenance tickets. These fault classifications were then linked to the sub-systems. However, no decisive conclusion could be drawn on the root cause or causes for these sub-systems' failure. Therefore, in this chapter, maintenance log analysis will take place. However, due to the complexity of this multi-component system, it is impossible to deduct the maintenance log analysis and implement the whole bridge in the proposed architecture of chapter 4 due to time constraints. Therefore, only the canopy will be analyzed to prove the proposed CPS architecture. Assumptions when implementing the proposed architecture are:

- The maintenance tickets provided by VolkerInfra are assumed to be correct and filled in truthfully.
- The behavior of the PBBs equipped with extra sensors is the same for all other Tianda PBBs when encountering the same type of aircraft.
- The components placed in the PBBs equipped with extra sensors are the same as the components fulfilling the same purpose in the other Tianda PBBs.
- The model assumes that the reliability of the canopy and its sub-systems, before calculations are done, is close to one.

5.2. Objective & KPI's

The research goal of this research is to develop a prediction model to forecast an impending failure of the PBB to prevent downtime of the PBB during in-time use. This forecast must then result in maintenance activities of the PBB being done proactively, or a real-time gate switching could be suggested. All this together must lead to a decrease in the time delay of the turnaround process of the aircraft, which is currently directly affected if failure of the PBB occurs. This means that to show the impact of the model and thus indicate how the proposed PdM strategy is beneficial concerning the current situation, the CPS architecture must provide the following output:

In the lower-level model, the health status of the different sub-systems is determined based on various failure mechanisms. Here, the output of the lower-level model will be the health status of the sub-systems. The reliability of the sub-systems will be used to show the impact of the model and compare it to the current situation. The research defines reliability as "the ability of a system or component to perform its required functions under specified conditions for a period of time" (Wang et al., 2016). The definition of failure in this research is: "the event or inoperable state, in which any item or part of an item does not, or would not, perform as previously specified." Reliability can also be stated as:

$$R(t) = 1 - F(t) \quad with\ F(t)\ the\ probability\ of\ failure \tag{5.1}$$

Here, the cumulative probability of the failure is the probability the sub-system or system will fail over time. As the different sub-systems have different failure mechanisms, the probability density function can be different. Here, a conventional statistical distribution that can be used is the exponential distribution. Here, the reliability of the sub-system can be described as:

$$R(t) = e^{-\lambda * t} \tag{5.2}$$

Here λ can be specified as

$$\lambda = \frac{1}{MTBF} \tag{5.3}$$

Here, MTBF is the Mean Time Between Failures, which is used for repairable sub-systems; in this research, it is assumed that all sub-systems are repairable.

With the health status monitored and proactive maintenance being scheduled, the MTBF is expected to be enlarged. This then results in increased reliability. With the explanation of reliability as an indicator of performance, the PdM strategy should improve the MTBF. This means the maintenance moments must be planned on time before unexpected downtime occurs.

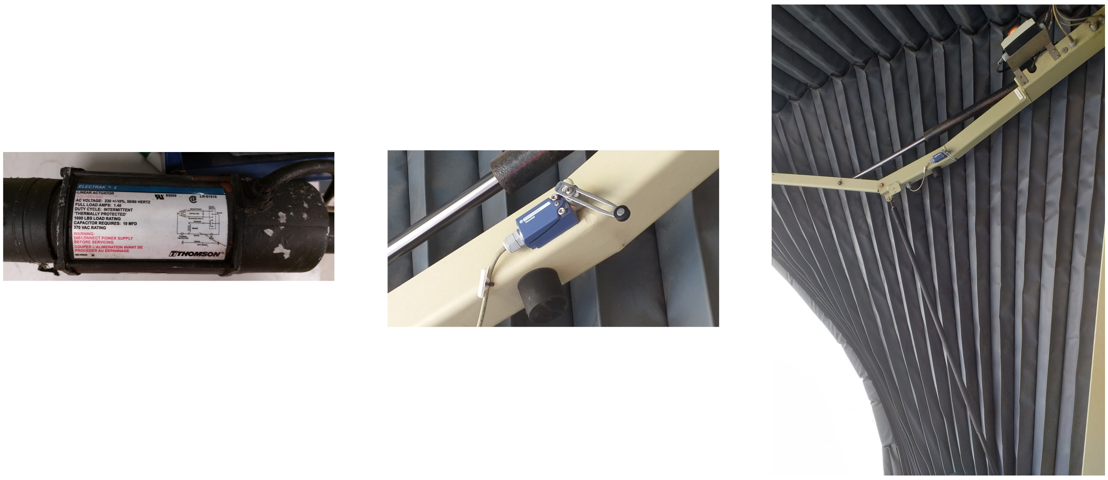
The second KPI is the maintenance repair time. Schiphol wants to have the availability of the aircraft stand as well as the PBB as high as possible. An inoperable PBB is unwanted, and fast repairments need to take place to have a functional PBB when the turnaround process starts. With the assumption that proactive repairments will be faster than corrective repairments, this KPI indicates that the model's output should be a planning of proactive maintenance moments.

The higher-level model should plan maintenance moments when the aircraft stand is not in use to reduce unexpected failures during in-time use. However, the availability of the aircraft stand is wanted to be as high as possible. Therefore, the third KPI is the repair rate. This means that the repair time over bridge uses is wanted to be as low as possible to execute maintenance activities.

5.3. Base case

To answer the following research question: "How does the developed PdM strategy perform in relation to the turnaround process?" a base case is defined. First, the sub-system, the canopy, must be defined. The canopy can be defined as a critical asset within the turnaround process. If the canopy fails, the turnaround process will be influenced directly, and delay will be imposed. The following reasons led to this conclusion: First, extending the canopy toward the plane is mandatory for some airlines. A failure of the canopy while extending towards the aircraft will lead to a safety issue, and a maintenance engineer must come. Second, a failure of the canopy while retracting results in a stop of auto parking on the bridge. Nevertheless, the bridge can go 1.5 meters back from the plane if the canopy fails halfway while moving in. However, a marshall needs to come to ensure everything is safe before the pushback of the aircraft can start. Third, in all cases above, a maintenance ticket will be made, and a maintenance engineer must come to the bridge to check the canopy and repair it if necessary.

Within the canopy, two segments are responsible for being able to extend or retract. One segment is for the left side of the canopy, and one is for the right side. The segments can be divided into the drive, the limit switches and the mechanical part. The three parts can be seen in figure 5.1.



(a) The canopy drive (b) The limit switch (c) The mechanical part

Figure 5.1: The components of the canopy

The canopy drive is a linear actuator. A limit switch is a mechanical device that requires the physical contact of an object with the switch's actuator to make the contact change state (open/closed). As the object or target contacts the operator of the switch, it eventually moves the actuator to the "limit," where the contacts change state. In a normally closed circuit, this mechanical action opens the electrical contacts and in a normally open circuit, it closes them. The contacts then start or stop the electrical circuit's current flow. Within the segment, there are two limit switches. The limit switch ensures that the canopy drive is shut down if the canopy is back at its starting position, and the limit switch ensures a stop of extending if the canopy is over the aircraft. The mechanical part consists of the beams that support the drive for extending and retracting and a gas spring. The gas spring ensures the canopy stays at the desired height during movement and when fully extracted.

From the maintenance tickets provided by VolkerInfra, the time between a failure is kept track of. The MTBF is calculated for the different failures of the canopy. The bathtub curve is used as a reliability tool to model the reliability of the canopy over time, and the assumption is made that the canopy is in the use-full life period, which means that the failure rate is constant. With this assumption, the exponential distribution is used to model the reliability of the canopy. The MTBF and reliability are calculated over the period 1 April 2019 - 31 December 2022 and indicated in table 5.1.

Table 5.1: The MTBF and failure rate of the canopy and its components

	MTBF (days)	Failure rate
Canopy	671.7	0.001488
Drive	858.6	0.001164
Limit switch	858.0	0.001165
Gas spring	895.1	0.001117

To get a distribution of the corrective repair time for each component, historical data is used, and the outliers are excluded from this research. With the use of the Python package fitter, the distribution for the drive and the gas spring is a gamma distribution. With a lognormal distribution for the limit switch.

The plots for the drive can be seen in figure 5.2; the limit switch and gas spring were done similarly.

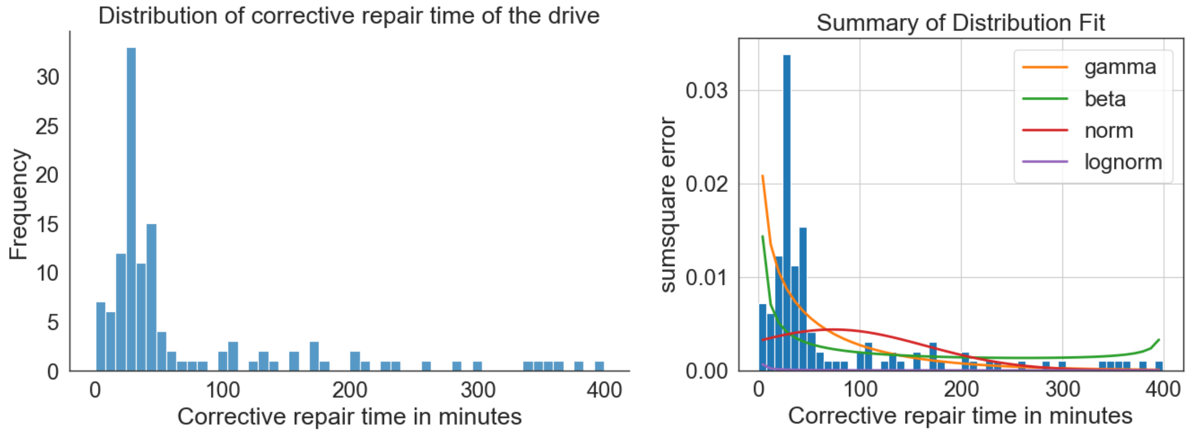


Figure 5.2: Drive repair time and distributions

The availability of the aircraft stand is determined via the data of D16, D18 and D51, assuming a normal distribution with a mean of 76 minutes with a standard definition of 54 minutes. As of now, the maintenance executed is unexpected downtime. Therefore, the maintenance moments during non-in-time use are only the preventive planned maintenance moments. According to the FMECA, the preventive measures are quarterly, half-yearly, and yearly, which means four preventive maintenance moments yearly.

Now the objectives and base case are defined, a simulation model can be developed to show how the architecture can be implemented on the PBB. From this implementation, the results of applying this model will show what the impact of the model will be. In the next sections, the lower-level model will be described first, followed by the higher-level model.

5.4. Lower-level model

In the lower-level model, the health status of the different sub-systems will be determined using the connection, conversion and cyber layer. With the canopy only in scope, the system role will shift from the PBB to the canopy. As for the canopy, the causes of failure are not clear yet; first, the preliminary layer, as described in section 4.2, will start with maintenance log analysis and gaining knowledge from experts.

The CLEAR Framework presented in Lo (2023) for prompt engineering will be used. The authors developed the CLEAR Framework to optimize the usage of LLM to enable the users to be more effective. The CLEAR Framework stands for Concise, Logical, Explicit, Adaptive, and Reflective. Being concise in constructing the prompt enables the language model to focus only on the parts for which an answer needs to be found. Removing unimportant information and being clear in the instructions will result in effective prompts. The second component of the framework is related to logical prompts. Logically constructing the prompt helps the AI to understand the relationship between the components of the prompt. This results in more precise and coherent output. For the third component, the prompt needs to be explicit in what the output structure needs to be. To get the desired output, the prompt needs to be specific about what kind of output format, scope, or content needs to be used. Flexibility is key while designing the prompt. Experimenting with different formulations or temperature settings is needed to find the optimal balance of the prompt regarding creativity and concentration. Thus, being adaptive in reformulating the prompts is needed in prompt engineering. The last framework component is related to being reflective in doing prompt engineering. Continuously evaluate the results and try to be critical towards the given output results to improve in prompt engineering (Lo, 2023).

For the maintenance log analysis, the maintenance tickets from 1 April 2019 to 31 December 2022 are used. In appendix E, the Python code used is displayed. To use only the relevant data for the log analysis, the data set has been reduced to only contain information about the classification and maintenance logs. Due to the structure of the data file, first, the four maintenance log columns need to be combined into one column, which captures all the information. Then, the names of the mechanics are removed from the logs to remove unnecessary information from the logs. Next, the column is filtered on each failure classification, and the analysis for them is done separately. Now, the CLEAR framework is used to design an effective prompt. This will result in a more structured way of designing the prompt and getting the desired output. First, what is needed from the AI needs to be determined. By filtering the canopy maintenance tickets in their specific fault classification, i.e., drive defect, limit switch defect and closing error, a first step in narrowing down the search has been done. For the drive defect, the interest is what caused the drive to fail. So, to be concise, we only need to ask the model to find the cause of the drive failure. Second, the prompt needs to be logical. Here, the model will be informed that each line contains a maintenance log where the cause of drive failure could be found. Next, the model will be informed that if no cause can be found, this must be stated. Thirdly, the output format needs to be known. Here, a decision must be made on display if dozens of logs will be analyzed.

From the results of using an LLM model, interesting results are seen. The model can provide a fast, structured answer using only a simple prompt. However, after implementing the model on larger parts of logs, a lot of hallucination in the output is seen. This can be solved by constantly evaluating the output and adapting to the situation. However, perfecting the prompt engineering is impossible in this research's time frame. Secondly, creating an LLM capable of analyzing the maintenance logs is not an objective of this research. AI is only used as a tool and gives a hopeful insight into the future. An intelligent way of using new technology is by creating a model in which large parts of maintenance logs can be analyzed and root cause analysis can be derived. The only uncertainty is the reliability of the model. Using AI LLM, the black box between input and output makes it hard to verify the model. It is unknown if the output is the actual output. Therefore, this way of maintenance log analysis can be questionable.

As maintenance log analysis could not provide the root causes of failure for well-considered data collection, expert knowledge was consulted. In section 2.4.2, it was seen that the canopy maintenance tickets were classified almost equally into three groups: failure related to the canopy drive, limit switches and an opening and closing. For the drive failure, the maintenance ticket analysis showed that 1/3 of the tickets could be assigned to this failure mechanism. From the maintenance logs, it was seen that the whole canopy drive then needs to be replaced. From expert knowledge, it became clear that not the drive as a whole but the torque limiter within the drive was the cause of failure. In general, torque limiters transmit the torque from the inner shaft to the output shaft. When a preset torque limit is exceeded, the two shafts slip with respect to each other to protect the parts of the drivetrain during overload. The torque transmitted during the overload depends on the torque limiter used and the situation occurring. The torque limiter used in the linear actuator of the canopy is a ball detent torque limiter. This type of torque limiter uses a series of balls placed in indentations. A spring force is applied to keep the balls in the indentation. If the preset torque limit is exceeded, the ball slips out of the groove and will slip until it falls back in the indentation.

To determine the health of the canopy, the drive can be monitored in three non-destructive ways: temperature, vibration and motor current (Hashemian, 2011). The first step was to start with monitoring the drive's motor current. This means that as a data source, the PLC data of the canopy is needed, and the sensors need to be installed to measure the current draw of the canopy drives. On 13 July 2023, sensors were installed to monitor the current draw through the canopy drives. The sensors are installed at aircraft stand D51, and the left canopy drive is measured. The choice for this stand is purely logistic based due to the constant use of stands D16 and D18 due to the summer holidays. Stand D51 is a narrow-body VOP. This means that only aircraft types with a diameter below 4 meters can park at this stand. Due to the almost identical shape of those narrow-body VOPs, the canopy's motion is assumed to be identical for each aircraft type; only the time can differ. Figure 5.3 shows the current overtime for extracting and retracting the left canopy segment. This profile is for fully extending and retracting without connecting to an aircraft. From figure 5.3, it can be seen that the current graph has an offset of 0.23 A for retracting and 0.29 A for extending.

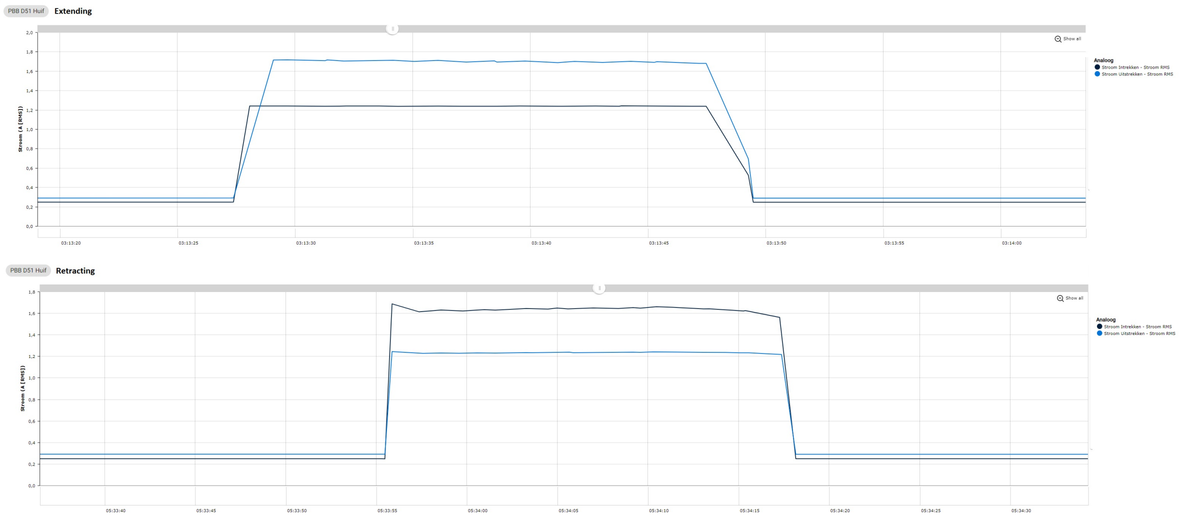


Figure 5.3: Canopy extending and retracting current over time

From table 5.2, the offset could be related to the current draw at no load. However, from expert knowledge, it was retrieved that no current is on the canopy when it is not in use. This offset could not be explained by the experts. Also, the current draw at full load is too high when checking table 5.2. From manual checks, it turned out that the current for extending and retracting is around 0.7 A. The scale error, however, due to measurement from the sensor could not be explained by the experts. Although anomaly detection could be implemented and a search for normal and faulty behavior could be done, it was decided that due to time constraints and the two unexplainable deviations, no further research was done into the relation of the current on the drive's health.

Table 5.2: Specifications of the canopy drive

Specifications	Value
Connection	Flying Leads
End Play, Max	1 mm
Feedback	No Feedback
Gear Reduction	10:1
IP Rating	IP55
Manual Override	No
Material	Copper; Steel; Zinc; Aluminum; Plastic
Motor Control	None
Stroke	609.6 mm
Protection, End-of-stroke	Clutch
Protection, Overload	Clutch
Screw Type	Ball
Max. static load	18000 N
Current draw no load	0.5 A
Current draw max. load	1.3 A
Max. dynamic load	4500 N
Voltage	230 Vac
Weight	8.5 kg

To prove the architecture assumptions were made to have the output of the lower-level model serve as input for the higher-level model:

- To prove the CPS architecture, the condition of the drive can be monitored by measuring the temperature, current and vibration of the drive
- As CPS architecture is verified using a simplified model, the sensors that provide information about the condition of the gas spring and the limit switch will be out of scope as this will be modeled in the same way as the drive health indicators.

- Due to the lack of data to directly estimate the drive's health status based on the sensors, an alternative approach is used. The sensors are considered nodes within the BN to demonstrate their influence on the overall health state of the drive.

5.5. Higher-level model

Using a simplified lower-level input, the higher-level model can be modeled. The higher-level model consists of building and verifying the cognition layer, the BN and DBN, and the configuration layer, the decision-making code. As the cognition layer output is used as input for the configuration layer, it was built and verified first before starting with the configuration layer.

5.5.1. Bayesian Network

For the DBN, first, the BN needs to be specified. The procedure in Schietekat et al. (2016) is used to build the BN. The three questions that need to be answered are:

- What is the graph structure, the variables and their values/states?
- What are the parameters?
- What inference modes will be used?

The model is built in the software package GeNIe 4.0 Academic (BayesFusion, 2023).

.

As the model structure should be the physical representation of the variable and the state, the variables used in the BN are defined. Table 5.3 displays the variables' descriptions. The states of the sensor nodes Temperature, Current and Vibration are Normal (N) and Abnormal (AN). The states of the component and sub-system nodes are Working (W) and Failure (F). For the life-cycle node, the states are Low, Medium and High.

Table 5.3: The variable definitions

Variable	Description
Canopy	The health status of the canopy assuming the health status of the canopy drive, gas spring and temperature.
Gas spring	The health status of the canopy gas spring.
Drive	The health status of the canopy drive, assuming the state of the temperature, the current, and the vibration of the drive.
Limit switch	The health status of the canopy limit switch.
Life cycles	The life cycles of the canopy.
Temperature	The temperature of the canopy drive in Celsius.
Current	The current of the canopy drive in Ampere.
Vibration	The vibration of the canopy drive in Hertz.

Next up is the values of the variables. The variables used in this model are discrete. The variables temperature, current, drive, gas spring, life cycles, and limit switch are input variables, while drive and canopy are output variables. The description of the reasoning for this model structure is cause-effect. The probabilities used in the model are described in table 5.4. For the sensor nodes Temperature, Current and Vibration, it is assumed that the sensor readings are in the normal operation region and a small prior failure probability is present. This translates into the prior probabilities of 0.999 for a normal operating state and a prior probability of 0.001 for an abnormal operating state.

For the sub-system nodes Limit switch and Gas spring, it is assumed that their prior probabilities for failing are small. This results in a prior probability of working of 0.99 and for failing 0.01.

The conditional probability for the drive node based on the given state of the sensor node is determined by expert knowledge and domain knowledge. For the temperature node, it is known that if the temperature reaches abnormal regions, the drive automatically switches off due to a thermal switch. This means that the probability of the failure state of the drive is high, given the state of current and vibration in a normal operation state and the temperature in an abnormal operation state. To incorporate an uncertainty measure, a failure probability of 0.99 is used. For the current sensor, a failure probability of 0.2 is used if it is in abnormal regions and the vibration and temperature are in a normal operation state. The torque limiter must ensure the drive will not fail or be damaged. However, it was seen that

Table 5.4: Probability distribution requirements and descriptions

Variable	**Probability Description**
Canopy	P(canopy | gas spring, drive, limit switch, life cycles)
	The chance that the canopy fails given evidence of the input variables
Gas spring	P(gas spring)
	The chance that the gas spring fails
Drive	P(drive | temperature, current, vibration)
	The chance that the drive fails given evidence of the input variables
Limit switch	P(limit switch)
	The chance that the limit switch fails
Life cycles	P(life cycles)
	The chance that the life cycles fall within a specific state
Temperature	P(temperature)
	The chance that the temperature is in normal state
Current	P(current)
	The chance that the current is in normal state
Vibration	P(vibration)
	The chance that the vibration is in normal state

the quality of the torque limiter is not always as it is supposed to be. For the failure probability given a failure of the vibration sensor, the failure probability is also 0.99. For the combination of temperature and or vibration with current given in abnormal readings, the failure probability of 0.99 is used.

The conditional probability for the canopy is based on the state of the different sub-system nodes. Here, it is assumed that if the drive and gas spring fail, the canopy fails. For the limit switch, it is assumed that there is a 90% chance that the canopy will still function in low life cycles as the canopy shuts down after 28 seconds while extracting takes approximately 21 seconds. This results in the canopy being pushed into the wall for only 7 seconds, which will not directly result in the failure of the canopy when it has a low life cycle. When more life cycles pass, the probability of failure increases to 30 percent in the state Medium and 50 percent in the state High.

For the inference technique used in this model, the health status over time, based on evidence, is needed. This means that forward inference is used. In GeNIe, the clustering algorithm is used as default.

5.5.2. Verification Bayesian Network

With verification of the model, the question of if the model is right is answered. This results in that it must be checked if the model is built correctly and works properly. Verifying the model is an ongoing process as the model is developed over time (Sargent, 2011).

To verify the Bayesian network, the structure must be according to table 5.3, and the arcs must represent the cause-effect relationship. In figure 5.4, the structure, including the arcs, is visualized. Here, it is also seen that no loops are in the network, so the network is acyclic. The probabilities agree with table 5.4. Second, to ensure that the conditional independence of the nodes is correctly tested, tests have been executed with each parameter, and no influence on the conditional independence nodes is seen. Third, the probability of the drive in the state working (P(D=W)) is calculated and checked with the output of the GeNIe model to see if the model performs as expected.

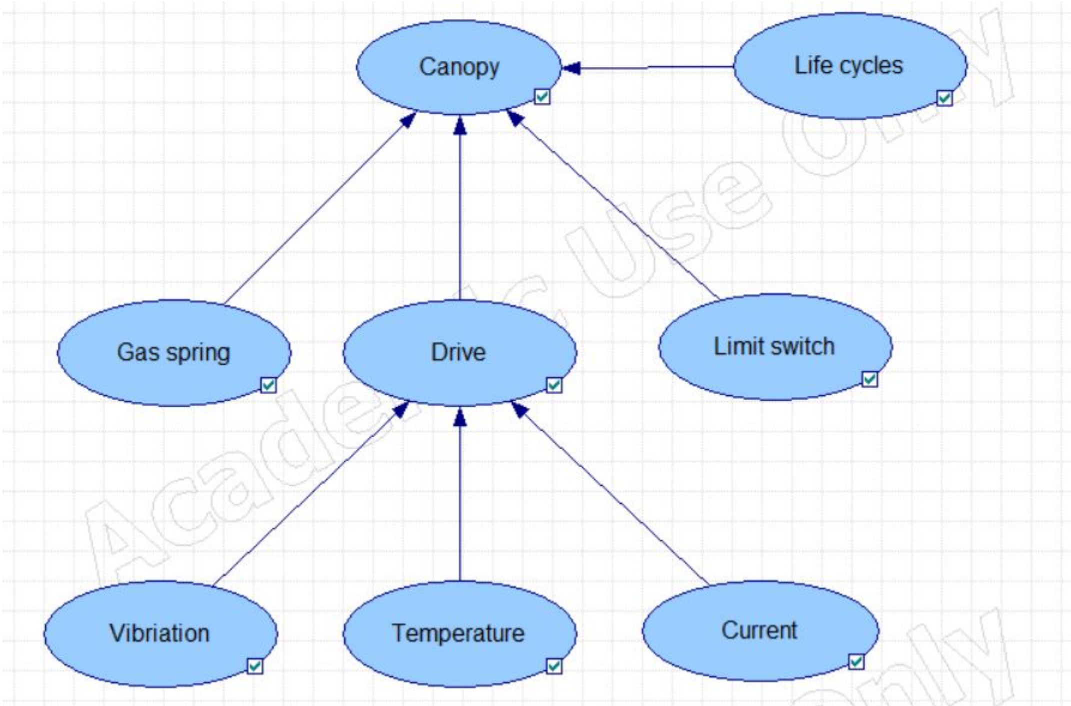


Figure 5.4: The Bayesian Network used for verification

$$\begin{aligned}
P(Drive = Working) = {} & P(Drive = Working, Vibration = Normal, Current = Normal, Temperature = Normal) \\
& + P(Drive = Working, Vibration = Normal, Current = Normal, Temperature = Abnormal) \\
& + P(Drive = Working, Vibration = Normal, Current = Abnormal, Temperature = Normal) \\
& + P(Drive = Working, Vibration = Abnormal, Current = Normal, Temperature = Normal) \\
& + P(Drive = Working, Vibration = Normal, Current = Abnormal, Temperature = Abnormal) \\
& + P(Drive = Working, Vibration = Abnormal, Current = Normal, Temperature = Abnormal) \\
& + P(Drive = Working, Vibration = Abnormal, Current = Abnormal, Temperature = Normal) \\
& + P(Drive = Working, Vibration = Abnormal, Current = Abnormal, Temperature = Abnormal)
\end{aligned}$$

as

$$\begin{aligned}
P(D = W) = {} & P(D = W, V = N, C = N, T = N) \\
& + P(D = W, V = N, C = N, T = AN) \\
& + P(D = W, V = N, C = AN, T = N) \\
& + P(D = W, V = AN, C = N, T = N) \\
& + P(D = W, V = N, C = AN, T = AN) \\
& + P(D = W, V = AN, C = N, T = AN) \\
& + P(D = W, V = AN, C = AN, T = N) \\
& + P(D = W, V = AN, C = AN, T = AN)
\end{aligned}$$

with

$$
\begin{aligned}
P(D = W, V = N, C = N, T = N) &= P(D = W|V = N, C = N, T = N) \cdot P(V = N) \cdot P(C = N) \cdot P(T = N) \\
&= 0.99 \cdot 0.999 \cdot 0.999 \cdot 0.999 \\
&= 0.987032969 \\
P(D = W, V = N, C = N, T = AN) &= P(D = W|V = N, C = N, T = AN) \cdot P(V = N) \cdot P(C = N) \cdot P(T = AN) \\
&= 0.01 \cdot 0.999 \cdot 0.999 \cdot 0.001 \\
&= 0.00000998 \\
P(D = W, V = N, C = AN, T = N) &= P(D = W|V = N, C = AN, T = N) \cdot P(V = N) \cdot P(C = AN) \cdot P(T = N) \\
&= 0.8 \cdot 0.999 \cdot 0.001 \cdot 0.999 \\
&= 0.0007984008 \\
P(D = W, V = AN, C = N, T = N) &= P(D = W|V = AN, C = N, T = N) \cdot P(V = AN) \cdot P(C = N) \cdot P(T = N) \\
&= 0.01 \cdot 0.001 \cdot 0.999 \cdot 0.999 \\
&= 0.00000998 \\
P(D = W, V = N, C = AN, T = AN) &= P(D = W|V = N, C = AN, T = AN) \cdot P(V = N) \cdot P(C = AN) \cdot P(T = AN) \\
&= 0.01 \cdot 0.999 \cdot 0.001 \cdot 0.001 \\
&= 9.99 \times 10^{-9} \\
P(D = W, V = AN, C = N, T = AN) &= P(D = W|V = AN, C = N, T = AN) \cdot P(V = AN) \cdot P(C = N) \cdot P(T = AN) \\
&= 0.01 \cdot 0.001 \cdot 0.999 \cdot 0.001 \\
&= 9.99 \times 10^{-9} \\
P(D = W, V = AN, C = AN, T = N) &= P(D = W|V = AN, C = AN, T = N) \cdot P(V = AN) \cdot P(C = AN) \cdot P(T = N) \\
&= 0.01 \cdot 0.001 \cdot 0.001 \cdot 0.999 \\
&= 9.99 \times 10^{-9} \\
P(D = W, V = AN, C = AN, T = AN) &= P(D = W|V = AN, C = AN, T = AN) \cdot P(V = AN) \cdot P(C = AN) \cdot P(T = AN) \\
&= 0.01 \cdot 0.001 \cdot 0.001 \cdot 0.001 \\
&= 1 \times 10^{-11}
\end{aligned}
$$

result in

$$P(Drive = Working) = 0.987851$$

which is the same as the output given in GeNIe.

At last, a sensitivity analysis of the network has been done to test the robustness of the BN, figure 5.5. The sensitivity analysis provided the impact on the canopy node if the underlying nodes have a 10% increase or decrease. The figure shows that the Canopy node is sensitive to a change in the drive node and its parent node, as the canopy fails directly if the drive fails. This also explains the sensitivity to the gas spring node. It is, therefore, key to ensure that implementing this architecture has accurate descriptions for these nodes. Based on the results, it is concluded that the BN is built correctly and can be used as a basis for the DBN.

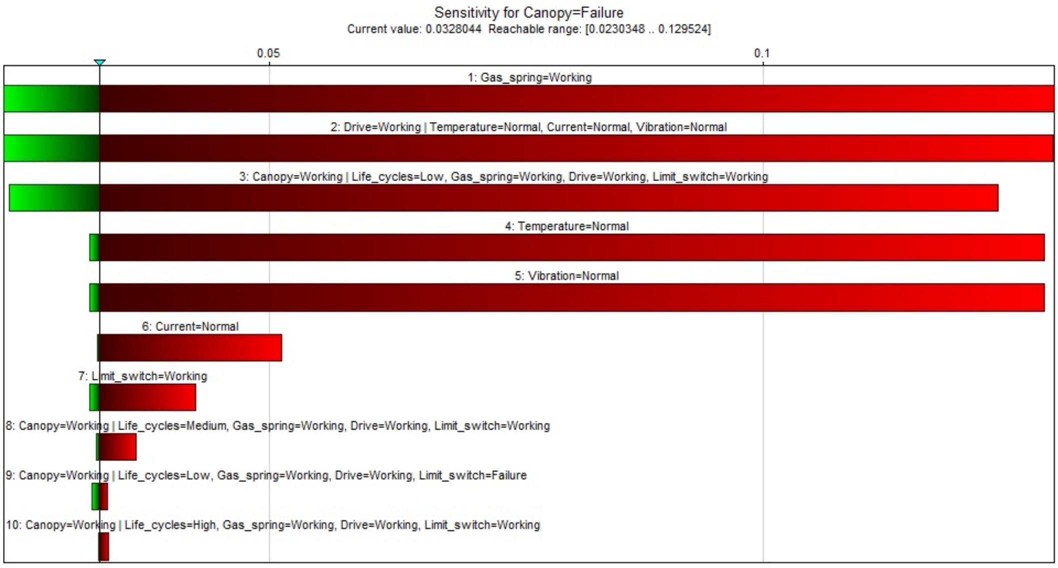


Figure 5.5: Sensitivity tornado of the Bayesian Network

5.5.3. Dynamic Bayesian Network

The input for the DBN is the BN combined with the transition of the states over time from the components and canopy. For the time step, one aircraft handling is assumed to happen at a time step. This means that the DBN predicts the health of the canopy for upcoming aircraft handling. The MTBF is calculated in the base case. As the amount of aircraft processed daily differs per aircraft stand, a mean of five has been chosen to prove the architecture. For the degradation over time, the MTBF is chosen. The transition probability is calculated with equation 5.4.

$$F(D = N|D = N(t-1)) = 1 - exp^{(-\lambda * \Delta t)} \quad with\ \lambda = 1/MTBF \tag{5.4}$$

Since the nodes of the sensors are evidence nodes, no temporal arcs are placed on the nodes. Also, in this model, it is assumed that the three components are solely responsible for the failure of the canopy. The resulting DBN is visualized in figure 5.6.

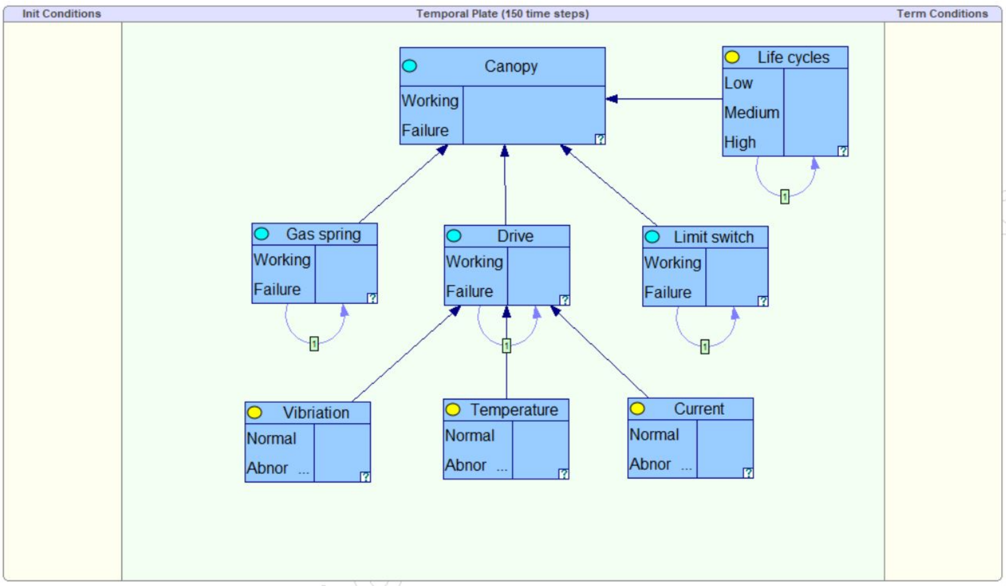


Figure 5.6: The Dynamic Bayesian Network used for verification

5.5.4. Verification Dynamic Bayesian Network

For the DBN, the underlying BN is already verified. Here, only the temporal nodes and the degradation over time provided evidence must be checked to see if they worked according to the expectations. As the current health situation of the canopy is not known beforehand, the failure probability should not start at zero, and it is also not seen. Further along, extreme cases in providing evidence are implemented to see if the expected output corresponds with the output provided by the model. By inserting evidence of failure in the nodes, all the nodes give failure as output, which is then correctly processed by the already verified BN.

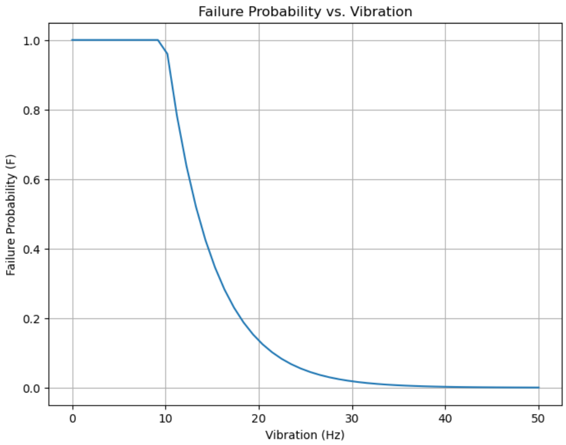
5.5.5. Decision-making

The output of the cognition level serves as input for the configuration layer. Here, decision-making will be based on the formulas presented in section 4.6. The input for this is:

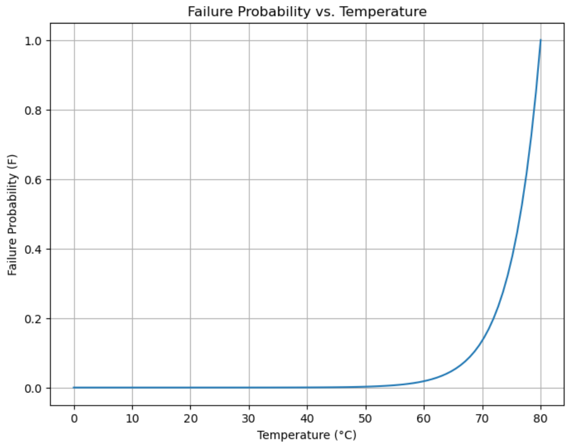
- Amount of components
- Failure probability components
- Corrective repair time components
- Proactive repair time components
- Arrival time
- Time step and period
- Availability aircraft stand
- Mechanics schedule
- Reliability
- Alpha
- Beta

The components used are the drive, limit switch and gas spring. The failure probabilities come directly from the output of the cognition layer and are used here as input. The corrective repair time is determined based on historical data from the maintenance tickets provided by Volker as presented in section 5.3. Here, the corrective repair for each fault is stated. A distribution is derived by plotting the different repair times over their frequency. The proactive repair time is set on half the corrective repair time to show the impact. The arrival time used in the decision-making model is also already defined in the base case. For the period, 450 bridge uses have been chosen. A longer time step was not chosen for predicting the future; there is too much uncertainty for a longer time frame. The same availability is used as in the base case. The number of mechanics is set to 1, and it is assumed that the number of mechanics does not change between timesteps. Alpha and beta are set to 1. The threshold for reliability can differ per sub-system or asset and purpose. The threshold in this research is set at 0.1 for the drive, the limit switch, and the gas spring. It was further decided that for the system canopy, no maintenance is executed if the reliability of the canopy is higher than 90 percent.

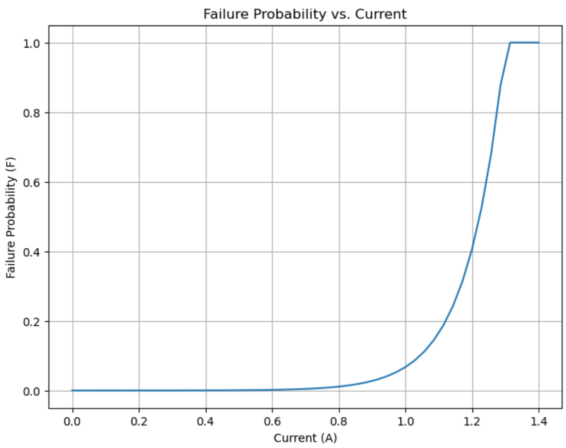
A synthetic data set will be used for the sensor readings of the first 100 aircraft handelings. With this dataset, the impact of condition monitoring on predicting the health status of the different components in the future will be shown. As the value of these readings varies during the in-time use of the PBB, it is impossible to use only one value for the whole duration of aircraft handling. However, to show the model's potential, a generic value will be used to show the model can predict an upcoming failure and that proactive maintenance will be scheduled. In figure 5.7, the relationship between the value of the sensor nodes and the corresponding failure probability can be seen. The range for the current and temperature sensor was based on the specifications of the canopy drive, for the range of the vibration sensor no characteristics could be found in the specifications of the canopy drive. Within vibration analysis, multiple techniques can be used to find the working frequency of the drive (Akbar et al., 2023), however, the goal of the synthetic dataset is to show the DBN can update its beliefs. This resulted in it being assumed that the drive has a mid-frequency range and all frequencies below 50 hz will indicate a failure of the drive. The assumption for all sensors was made that the failure probabilities can be related to an exponential distribution. The synthetic dataset was generated, appendix D, and the corresponding failure probabilities were calculated. Next, the failure probabilities were implemented in GeNIe. From here, the failure probability for the drive was recalculated.



(a) Vibration sensor



(b) Temperature sensor



(c) Current sensor

Figure 5.7: Sensor values with corresponding failure probability

To ensure the decision-making model is built correctly and works as intended, qualitative verification is used throughout building the model. A separate section in the notebook environment of Visual Studio Code has been used for each step of the decision-making process. In this way, the model is built correctly. The Python code can be found in Appendix F.

5.6. Test plan

With the CPS architecture used for the PBB, the impact of this proposed solution will be shown with the simulation model. The following tests will evaluate the KPIs of the base case versus the newly designed situation. First, the base case will be implemented, and the optimal time to do maintenance is shown. Second, the synthetic dataset is implemented, and a new prediction is presented. For both of the predictions, the influence of the proactive repair time will also be investigated. As for the base case, the proactive repair time is set at half the corrective repair time; in the test, the proactive repair time will vary between 10 percent and 90 percent of the corrective repair time. As the proactive repair time influences the expected repair rate of the system, it influences the outcome of the expected total repair rate. However, the availability of the aircraft stand to do maintenance is set as a constraint. It can thus be the case that due to a higher proactive repair time, the availability constraint of the aircraft stand is exceeded, and a new optimal maintenance moment needs to be found compared to a lower proactive repair time. This will also be done with the number of maintenance mechanics to show the impact on the availability versus the repair time and the repair rate.

5.7. Conclusion

This chapter partially implements the proposed CPS architecture for a multi-component system using a simulation model. First, the assumptions and boundaries were defined, followed by the base case. The base case is a reference to see how the proposed architecture relates to the current situation within the turnaround process. The lower-level model was applied to the PBB, focussing on the sub-system canopy. Due to time constraints and questions about the validity of the sensor data, a simplified output was used to serve as the output of the lower-level model. With the canopy implemented on the higher level model, the DBN was formulated and built to serve as input for the decision-making. The DBN was verified, and a sensitivity analysis was done to show its robustness. It was seen that the DBN was highly sensitive for the given prior probabilities. In the highest layer of the CPS, the decision-making process was implemented for the canopy. A test plan was defined to see how the output behaves in two scenarios. In the next chapter, the results from the test plan are presented.

6
Results

In this chapter, the results of the test plan introduced in section 5.6 will be elaborated. This chapter answers the sub-question: "How does the developed predictive maintenance strategy perform in relation to the turnaround process?". In the first section, the base case results are presented, followed by the results of the higher-level model based on the synthetic dataset. In the last sections, the influence of the height of the proactive repair time and the mechanics on the decision-making output are shown.

6.1. Base case

The base case is used together with the input as presented in section 5.3. The DBN model calculated the failure probabilities for the given period of 450 bridge uses. In figure 6.1, the failure probability of the canopy as a system is plotted based on the results from the DBN. In the figure, it can be seen that the failure probability crosses the reliability line, the dashed black line, at 23 bridge uses. As stated, no maintenance will be executed if the reliability of the canopy is above 0.9.

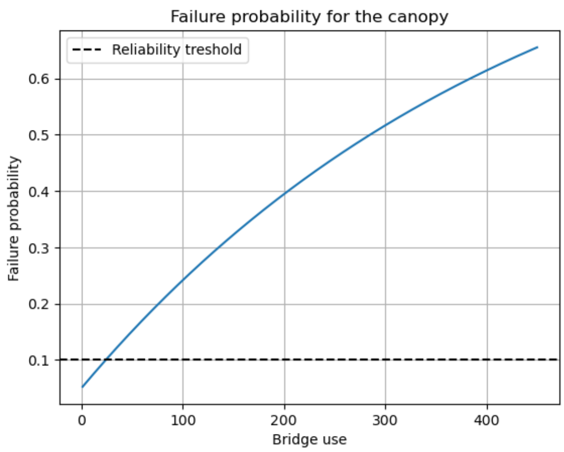


Figure 6.1: The failure probability of the canopy

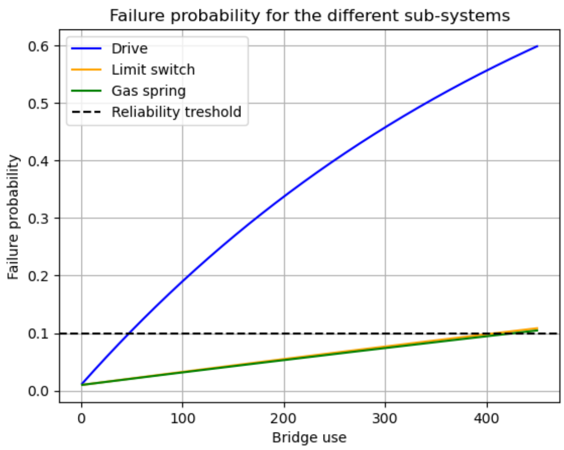


Figure 6.2: The failure probability of the sub-systems of the canopy

Now, the sub-systems of the canopy can be looked at. Figure 6.2 shows the failure probabilities of the different sub-systems, the components of the canopy: blue for the drive, orange for the limit switch and green for the gas spring. The reliability threshold, the dashed black line, was set at 0.9; the points where the failure probabilities exceed the threshold (1-0.9=0.1) are shown. The drive needs to be maintained before the 49th bridge use, the limit switch before the 410th bridge use and the gas spring before the 428th bridge use. The local opportunistic strategy searched for the optimal moment to do maintenance for each sub-system individually by choosing the maintenance moment where the expected total repair rate is at its lowest while respecting the availability and reliability threshold. To show that the decision model respects the availability constraint, the drive repair time is plotted as an example in figure 6.3, which is the expected proactive repair time plus the arrival time. The availability of the aircraft stand to do maintenance is plotted in light gray. It can be seen that the total repair time is less than the availability of the aircraft stand for the given timesteps except between bridge use 43 and 44. However, as plotted in figure 6.4, the expected total repair rate of the drive is at its lowest by doing maintenance between bridge uses 44 and 45. This moment is, therefore, chosen by the model for the optimal time to do maintenance. In table 6.1, the optimal time to perform maintenance is given together with the corresponding expected repair rate. For the drive, the optimal time to do maintenance is between the 44th and 45th bridge use, for the limit switch between the 400th and 401th bridge use, and for the gas spring between the 413th and 414th bridge use.

Table 6.1: The optimal maintenance moments and the expected total repair rate of the sub-systems

Sub-system	Optimal maintenance moment	The expected total repair rate
Drive	Between bridge use 44-45	0.384
Limit switch	Between bridge use 400-401	0.045
Gas spring	Between bridge use 413-414	0.050

When applying the global opportunistic strategy, the decision-making algorithm searches for a maintenance group if it reduces the expected minimal total maintenance rate. By combining this with the restriction that the combined repair time cannot exceed the availability of the aircraft stand, the search for an optimal maintenance group can be started. Due to the optimal maintenance moment for the drive and the limit switch to be more than 350 bridge uses away no optimal group could be found. However, as can be seen from table 6.1, the optimal time to do maintenance for the limit switch and gas spring are close to each other. Therefore, the maintenance decision code was slightly adapted to see if combining the maintenance moment for the limit switch and gas spring together is beneficial for the expected minimal total maintenance rate. From the results, it was seen that combining the limit switch and gas spring in a maintenance group will drop the expected minimal total maintenance rate from 0.479 to 0.453. Here the new optimal time to do maintenance for the group is between bridge use 395 and 396.

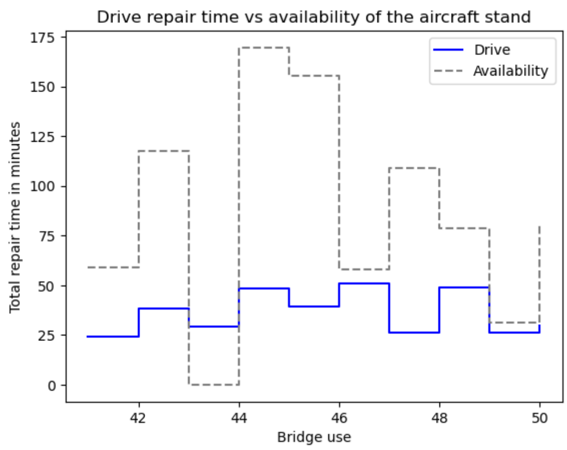


Figure 6.3: The availability of the aircraft stand versus the time needed for repairments

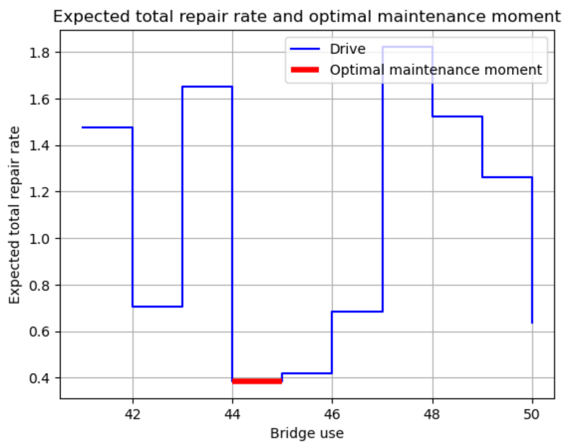


Figure 6.4: The expected total repair rate for the drive

6.2. Synthetic dataset case

The synthetic data set is used to show the ability of the model to adapt to new information and what the impact is on the failure probabilities, repair time, optimal maintenance time and expected repair rate. As the synthetic data set does not influence the results of the limit switch and gas spring, these figures and values are not included in this part of the results. In figure 6.5, the updated failure probability is shown. By implementing the sensor readings as evidence for the sensor nodes in the DBN, the time step at which the failure probability exceeds the threshold shifts. Maintenance of the drive must occur before the 112th bridge use to comply with the reliability threshold. Figure 6.5 shows the expected total repair rate of the drive in blue, with the optimal maintenance moment from the base case in black and the newly determined optimal moment to do maintenance in red. The optimal maintenance moment is shifted to between bridge use 108 and 109. Due to the model update based on evidence for every time step, the previous 100 bridge uses, the expected total repair rate of the drive at the optimal time to do maintenance dropped from 0.384 to 0.149. Although the optimal time to do maintenance for the drive shifted to between bridge use 108 and 109, the gap between the maintenance moments of the limit switch and gas spring was too big to form an optimal maintenance group in the global opportunistic strategy.

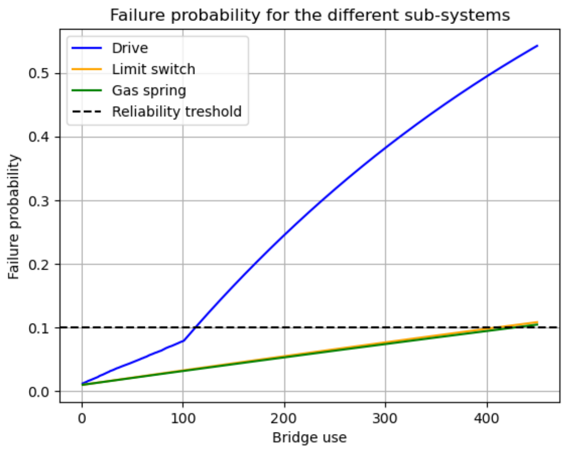


Figure 6.5: The failure probabilities when evidence is added

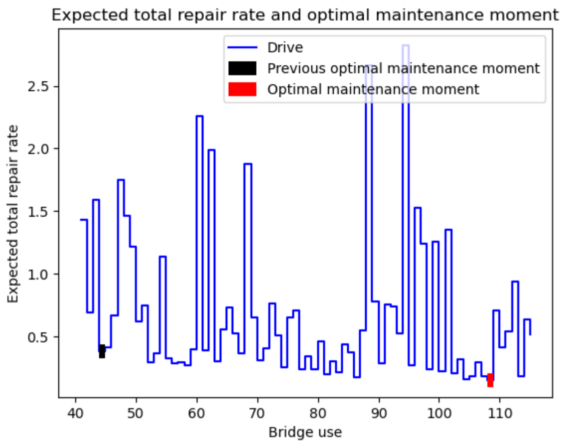


Figure 6.6: The total repair rate of the drive and the optimal maintenance moment

6.3. Influence of the proactive repair time on the decision-making

The proactive repair time was varied to show the impact on the model's output. The failure probabilities of the base case, presented in figure 6.2, are first used. Thereafter, the influence of the proactive repair time is presented for the synthetic dataset case. Important to notice is that in the base case and synthetic dataset results, as presented in the first two sections of this chapter, a proactive repair time of 0.5 times the corrective repair time is used.

6.3.1. Influence on the base case

For the base case, a range of values for the proactive repair time was implemented. From the outcome of the decision-making model, it was seen that the optimal maintenance moment was not shifted for the drive and limit switch when using different values of the proactive repair time. However, for the gas spring, the outcome showed that if the proactive repair time is higher than 0.6 times the corrective repair time, the optimal time to do maintenance is shifted to almost 100 bridge uses earlier, namely between bridge uses 336 and 337. In figure 6.7, the expected total repair rate for the gas spring is plotted. In black, the expected total repair rate of the gas spring for in-between bridge use 413 and 414, and in red, the expected total repair rate for in-between bridge use 336 and 337. In this graph, it can be seen that for a proactive repair time of 0.7 times the corrective repair time or higher, the total expected repair rate is lower for in-between bridge use 336-337 than the optimal maintenance time of in-between bridge use 413-414. As the model searches for the maintenance moment where the expected total repair rate of the sub-system is at its lowest, the optimal time to do maintenance for the gas spring is chosen as in between bridge use 336 and 337 for a proactive repair time of 0.7, 0.8 and 0.9 times the corrective repair time.

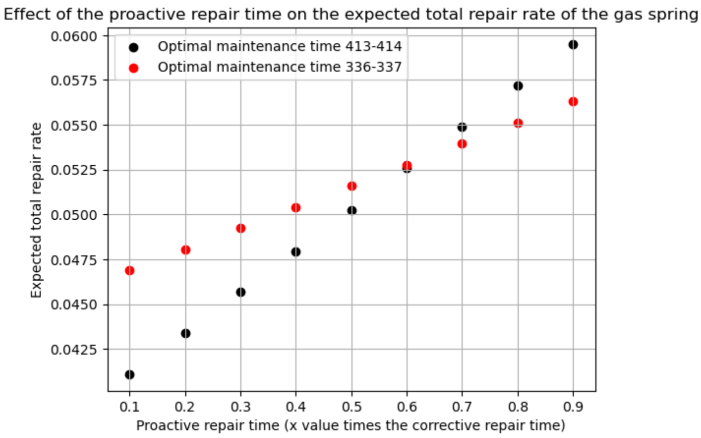


Figure 6.7: Effect of the proactive repair time on the expected total repair rate of the gas spring

In figure 6.8, the result of forming different maintenance groups is shown for the expected total repair rate of the system at the determined optimal time(s) to do maintenance. In purple, the result of forming a maintenance group as presented in the global opportunistic strategy procedure in 4.6 is shown. Comparing it with repairing all the sub-systems individually, indicated with black dots, it can be seen that for each value of the proactive repair time, it is more beneficial to maintain the sub-systems individually. However, as done in the base case, the limit switch and gas spring are combined into one maintenance group, the red dots in the figure. It can be seen that for each value of the proactive repair time, it is beneficial for the expected minimal total maintenance rate of the system to combine these two sub-systems in the maintenance group. The figure shows that from a proactive repair time of more than 0.6 times the corrective repair time, the expected total repair rate for the maintenance group according to the global opportunistic strategy is not linearly increasing for each group. This is because the optimal time for maintenance for the gas spring is shifted towards bridge use 336 and 337. This results in that the maintenance group formed is not the drive and limit switch but the drive and gas spring, which will give a different result in the expected minimal total maintenance rate for the system.

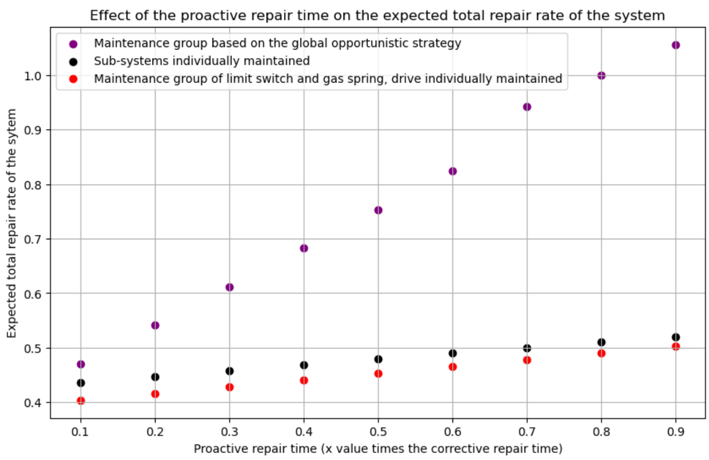


Figure 6.8: Effect of the proactive repair time on the expected minimal total maintenance rate of the system

6.3.2. Influence on the synthetic dataset case

The range of different values for the proactive repair time was applied in combination with the synthetic dataset for the sensor readings of the drive. For the local opportunistic strategy for the drive, the proactive repair time does not influence the optimal maintenance time, only the expected total repair rate. As the proactive repair time increases, the expected total repair rate increases. The optimal time to do maintenance for the drive is still between the 108th bridge use and the 109th. In figure 6.9, the result of forming different maintenance groups is shown for the expected minimal total maintenance rate of the system at the determined optimal time(s) to do maintenance, now updated with the synthetic dataset. In this figure, it can be seen that the optimal moment for maintenance for the gas spring is shifted to in-between bridge use 336 and 337. With a proactive repair time higher than 0.6 times the corrective repair time, the gap between the optimal maintenance moment for the drive, between bridge use 108 and 109, is decreased, and the expected total repair rate drops significantly. However, maintaining the sub-systems individually or a maintenance group of the gas spring and the limit switch is more beneficial for the expected minimal total maintenance rate of the system.

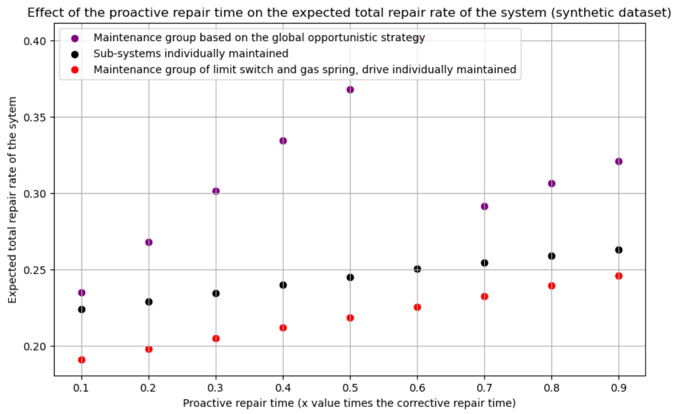


Figure 6.9: Effect of the proactive repair time on the expected minimal total maintenance rate of the system when the synthetic dataset is used

6.4. Influence of the mechanics on the decision-making

In this section, the number of mechanics varies to see its influence on the decision model's output. In this first section, the number of mechanics varies for the base case. In the second section, the number of mechanics varies for the synthetic dataset case. Important to notice is that in the previous sections, the number of mechanics was one.

6.4.1. Influence on the base case

The amount of mechanics was varied to show the impact on the model's output. The failure probabilities of the base case, presented in figure 6.2, are first used. With more mechanics and the assumption that ω_{CRT} and ω_{PRT} are one, the repair time of the sub-systems drops. Although the repair time is decreased by having more mechanics, the optimal time to do maintenance for the different sub-systems is not changed. Figure 6.10 shows the expected total repair rate of the different sub-systems. From the figure, it can be seen that by increasing the mechanics, the expected total repair rate for the sub-systems also drops. Having two mechanics repairing a sub-system, the repair rate drops 5.4 percent compared with one mechanic for the drive to almost 14 percent for the gas spring. When applying the global opportunistic strategy, although increasing the mechanics, no optimal maintenance group could be formed due to the early maintenance of the drive. The same as in the base case, the maintenance group of the limit switch and the gas spring is made, and the expected minimal total maintenance rate of the system is plotted against the expected minimal total maintenance rate of the system when maintaining the sub-systems individually. Also, for this maintenance group, the optimal time to do maintenance is unchanged and stays between bridge use 395 and 396. By both graphs, visualized in figure 6.11, the expected total repair rate will drop significantly by having two maintenance mechanics instead of one.

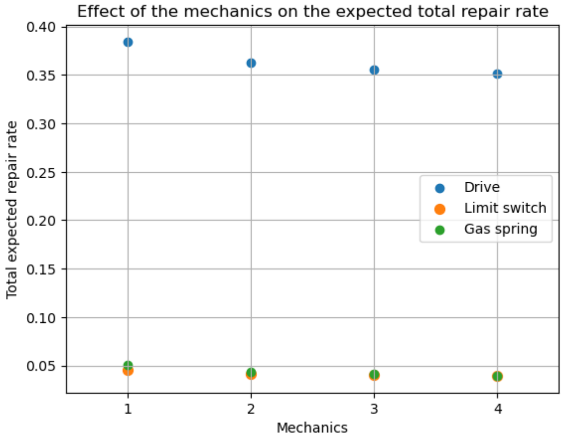


Figure 6.10: The effect of the mechanics on the total repair rate of the different components

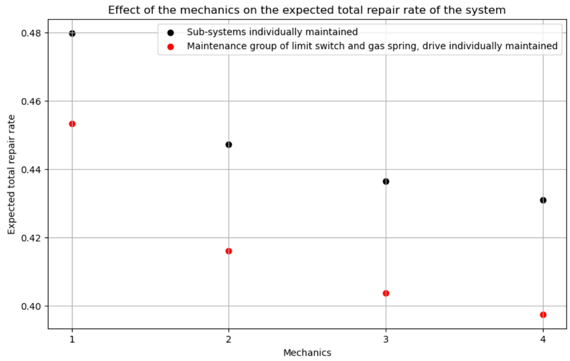


Figure 6.11: The effect of the mechanics on the expected minimal total maintenance rate of the system based on the formed optimal maintenance group

6.4.2. Influence on the synthetic dataset case

The number of maintenance mechanics was also varied in combination with the synthetic dataset for the sensor readings of the drive. The same as in the proactive repair time experiments, the number of mechanics varied in combination with the synthetic dataset did not influence the optimal time to do maintenance. When looking at the global opportunistic strategy, the gap is still too big to form an optimal group with the drive and the limit switch, as shown in figure 6.12. Here, the global opportunistic strategy maintenance group is plotted in purple, and the individual maintenance system is plotted in black. For each number of maintenance mechanics, the global opportunistic strategy gives a higher expected minimal total maintenance rate of the system than combining the drive and limit switch in a maintenance group while maintaining the gas spring individually. What is seen is a steep decrease in the expected repair rate for the system if the global opportunistic strategy is used, a drop of 25 percent. However, now the optimal time to do maintenance needs to take place before the 112 bridge use, which is almost 250 bridge uses earlier than if the limit switch is maintained individually. Also, the limit switch and gas spring are combined in a maintenance group to see if this is beneficial for the expected total repair rate of the system. From the red points in figure 6.12, it can be seen that having more mechanics and combining the drive and gas spring in a maintenance group is beneficial for the expected minimal total maintenance rate of the system.

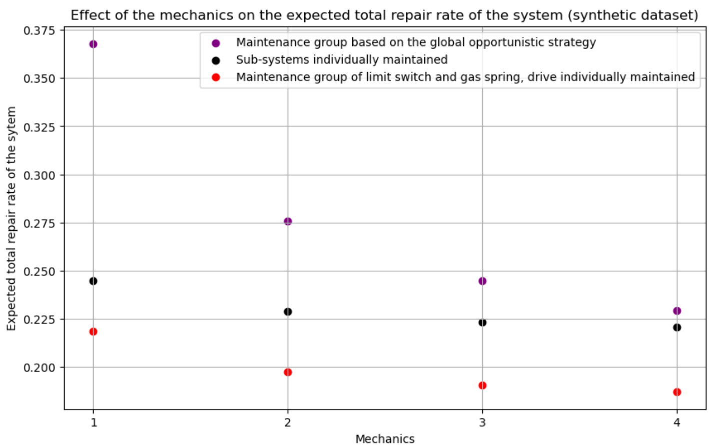


Figure 6.12: Effect of the number of mechanics on the expected minimal total maintenance rate of the system when the synthetic dataset is used

6.5. Results conclusions

In this section, the output of the higher-level model was shown. First, the base case input was used to determine the optimal maintenance moment for the sub-systems individually. After that, the global opportunistic model was used to determine the optimal maintenance group together with the group's optimal maintenance moment. It was shown that the model can determine the optimal maintenance moment for the component individually and for the optimal maintenance group by finding the lowest repair rate while respecting the availability constraints. It was seen that the procedure to find the optimal maintenance group by combining the first and second sub-systems placed for maintenance was, in this case, not beneficial for the system as the time between them was too long. As the optimal time to do maintenance of the limit switch and gas spring was only 13 bridge uses away, it was seen that if these two sub-systems were combined in a maintenance group, the expected minimal total maintenance rate of the system would be lower than maintaining these sub-systems individually. By implementing the synthetic dataset in the cognition layer, the DBN updates the failure probability of the drive and a new decision-making of the optimal maintenance moment is made. The results show that the higher-level model can update the model's reliability and choose the maintenance moment with the lowest repair rate while respecting the availability constraint. By determining this optimal maintenance moment, the proactive repair time can be used, which is less than the corrective repair time. This proactive repair time is also investigated in terms of how it influences the optimal maintenance moment and repair rate in the base and synthetic dataset case. The results showed that the height of the proactive repair time can influence the outcome of the optimal time to do maintenance for the gas spring if a proactive repair time of 0.7 or higher times the corrective repair time is used. For the global opportunistic model, the height of the proactive repair time only influenced the outcome for a proactive repair time as the order of maintenance is shifted due to the earlier optimal time to do maintenance for the gas spring. However, it did not change that it was not beneficial to use the global opportunistic strategy to form a maintenance group of the first and second sub-systems that require maintenance. The influence of the amount of mechanics on the repair rate was significant; it was seen that increasing the mechanics from only one to two already decreased the total repair rate from 5 to 14 percent for the local strategy of the gas spring and 25 percent for the global strategy in the synthetic case. However, it did not change the optimal maintenance moment.

7 Discussion

In this chapter, the results of this research are interpreted and discussed. This research proposes a CPS architecture for developing a PdM strategy for a multi-component system. By developing this architecture and running a simulation model, the impact of the model can be evaluated. In the following sections, each layer of the CPS for the multi-component system, the PBB, is discussed. In this research, validation and implementation were not reached, resulting in the thesis only focusing on the cyber part of the CPS. Although suggestions for maintenance activities were made in the configuration layer, a complete loop was not met.

7.1. Connection layer

The physical system is connected to the cyber part in the connection layer by collecting data from various sources. In the first instance, the ability to collect data was thought to be simple. However, collecting data without knowing what to predict was inefficient and time-consuming. A more concise data collection could be present by including a preliminary layer where the root causes of failure were first found. Within the preliminary layer, using maintenance log analysis and expert knowledge, the root causes of failure could be found. Within the research, the assumption was made that the maintenance tickets were correct and filled in truthfully. This assumption was made because manually correcting the maintenance tickets was not feasible within this research time frame and was not a goal of this research. Nevertheless, ensuring the data source is correct and truthful is crucial. As data can be collected from almost all sources due to the evolution of Industry 4.0, it is a waste of time and money if (sub)systems are monitored due to wrongly filled-in maintenance tickets and in the end, it is concluded that no faults are happening. For the canopy, the maintenance tickets showed that the general classifications were insufficient to find the root causes of failure and the maintenance logs were analyzed. With the use of the LLM, the output showed that the maintenance tickets logs were too generally filled in, and the text log, as how it was supposed to function, could not give the cause of failure and how it was solved. As maintenance log analysis was promising but insufficient, expert knowledge was needed. Here, it was seen that when implementing a theoretical architecture in practice, the mechanics working daily with the PBB are essential. This resulted in the canopy failure's root causes were found with the help of expert knowledge. In this phase, especially, the collection of data and the execution of the preliminary layer took more time than expected. Collecting data from the installed current sensor took much time and resulted in doubts about whether it was meaningful. However, this layer and the newly defined preliminary layer showed that the practical setting and the theoretical situation are not applicable one-to-one. In reality, significant data collection could not be as easy as expected.

7.2. Conversion layer

The conversion layer converts data from the connection layer to information. Only the data from the installed current sensors were used in the simulation. With this data, a first step was made to convert the data to information regarding the health status of the canopy. Due to the time constraint and uncertainty in the validity of the data, no further research was done on the current data and assumptions were made to use a simplified lower model output. Therefore, this layer was not proven in this research. However, as various research in PHM systems based on sensor data can be found in the literature, the importance of converting raw data to information can not be neglected.

7.3. Cyber layer

In the cyber layer, the different health statuses of the components come together to predict the sub-system health. Due to the complexity and size of the PBB within the given time frame, the canopy was used as a system, with the three components as sub-systems and life cycles as separate failure mechanisms. Due to time constraints, no further research was done on the sub-systems limit switch and gas spring. As the sub-system drive was started with three failure mechanisms, only one model was used in the end due to time constraints. However, if we take one step back, if it is looked at as a system as a whole, the canopy as a sub-system has three failure mechanisms, with all their own models to assess the influence of the sub-system canopy health. The proposed solution of using a multi-model approach by combining the data from the proposed sensors is reducing the complexity instead of using three different data-driven models. If combined with that, each sub-system uses a multi-model approach instead of individual models. The purpose of using a multi-model approach to reduce the complexity in this layer can be said to be proven.

7.4. Cognition layer

The cognition layer serves as the medium for visualizing the whole system. In this layer, the system as a whole is evaluated. The DBN was chosen to have a clear overview of the multi-component system and visualize the dependencies between different components. From the implemented simulation model, it was found that the DBN and BN can indicate and visualize the multi-component system in an easy-to-understand way. This reduces the complexity of the system. With the DBN, a prediction can be made on the failure probability of the different sub-systems and systems. From the test with a synthetic dataset, the model provides an updated DBN for 100 bridge uses and can update the failure probabilities. With the DBN, this layer can provide insight into the system's reliability and corresponding sub-systems. The sensor readings were randomly chosen as input for the synthetic dataset, and one value was used for the whole bridge use. This is not the case in reality. However, in this research, the sensors were chosen to be modeled as nodes to show their influence. When applying this architecture, the sensor nodes are replaced with the health status based on the sensor output, and this assumption is overcome. From the sensitivity analysis, it was found that the outcome of the DBN is highly dependable on the prior probabilities and the CPTs for the child nodes. By having good insight into these probabilities, an accurate description of the real world can be made. Unfortunately, the model could only be implemented partly and with assumptions. This results in a synthetic representation of reality. However, this layer proves the importance of visualizing the system and the need for an easy-to-understand system representation. The DBN can indicate and visualize the multi-component system. However, due to the importance of having correct values of the conditional probabilities of the DBN, the choice to use it can be questionable.

7.5. Configuration layer

In the configuration layer, decision-making is conducted for the system. Based on the predicted failure probabilities from the cognition layer, the model can calculate the optimal maintenance moment with and without evidence. The opportunistic PdM strategy based on cost was reformed to an availability objective. This resulted in a rate where repair time over bridge use was minimized to find the optimal moment to do maintenance. Respecting the reliability threshold and the availability of the aircraft stand to do maintenance, the local optimal moments to do maintenance were found. As the height of the input values was respectively low, the expected total repair rate dropped fast to close to zero. The model can form an optimal maintenance group for the global opportunistic strategy. However, as the optimal maintenance moments for the limit switch and gas spring are almost 350 bridge uses further in the period, forming maintenance groups is not ideal if the individual moments are not close. In this result, a group of only the limit switch and gas spring and doing maintenance on the drive individually could have suited better. When applying the decision-making model to the synthetic dataset, the result shows that the optimal maintenance moment is shifted to a later bridge use. Due to the reliability threshold, the drive needs maintenance within 12 bridge uses, and the optimal time to do maintenance is thereafter calculated as between the 107th and 108th. In this model, it is assumed that maintenance can be planned and executed immediately. In reality, this does not have to be the case. In the tests for the decision-making model, the value of the proactive repair time was also changed over a range of values times the corrective repair time. Here, the range of proactive repair time was applied to the base case and the synthetic dataset case. Here, it was seen that the value of the proactive repair time influences the outcome. Therefore, the exact value of the proactive repair time must be known. The amount of mechanics was also varied to see the effect on the decision-making model output. Here, mechanics increased from one to two, showed a significant decrease in the expected repair rate. However, the assumption was that if two mechanics worked on the same failure, it would reduce the repair time in half. This will not always be the case in reality.

8

Conclusion and recommendations

In this section, the research question of this thesis is answered. In section 8.2, the limitations of this research are stated, and a recommendation for further research is made. In the last section, the recommendations for Schiphol are listed.

8.1. Conclusion

In this thesis, research was conducted to find an answer to the following research question:
How to realize a predictive maintenance strategy for Passenger Boarding Bridges to benefit the airport's turnaround process?

To answer the research question, the research sub-questions need to be answered. The first sub-question was: *What are the failure mechanisms of the Passenger Boarding Bridge?* The failure mechanisms of the PBB were captured in the maintenance tickets of VolkerInfra. According to these tickets, it became clear that degradation of the system based on life cycles was not the primary cause of failure. Here, a possible explanation could be well-executed preventive maintenance activities or that the in-use time of PBB is short. With the provided maintenance tickets, the failure classifications already determined by VolkerInfra could be used. It was concluded that preset classifications were too general to derive the root cause of failure. The classifications only gave the symptoms of failure, for example, system jam or failure while operating, instead of the root cause of failure, which prevents the start of adequate monitoring to predict the failure from happening. With expert knowledge, the failure mechanism of one sub-system of the PBB could be found, the failure mechanisms of the canopy. From here, the sub-system was investigated to find the root cause of failure. For the canopy, the torque limiter played a significant role in causing the maintenance tickets.

The second sub-question was: *What is the current state of maintenance activities of the Passenger Boarding Bridges?* Firstly, reactive maintenance is executed when a PBB failure is reported through a maintenance ticket. The maintenance mechanic will be sent to the PBB; if necessary, repairs are executed to put the PBB back in use. Second, preventive maintenance is executed based on the FMECA. Here, quarterly, half-yearly and yearly inspections are done. In addition, extra sensors were installed to collect data and get insight into the PBB condition. However, no arguments were documented as to why the specific extra sensors were installed and why other data from the PBB were excluded from the monitoring. This results in that without having a clear overview of the root causes of failure, predictions made based on the monitoring data could be impossible because no failure captured would be present within the data.

The third sub-question was: *What are the state-of-the-art techniques regarding predictive maintenance?* The development of PdM for multi-component systems is still in an early phase; nevertheless, with the rise of Industry 4.0, the developments captured within will enhance the research of PdM for multi-component systems. The use of data-driven methods in combination with Big Data enables the monitoring of complex systems. Using a multi-model approach, this increasing complexity, as the system will have more components, can be countered and made accessible and comprehensible. Combining this with a well-defined architecture, for instance, the CPS, the relationships and dependencies of the components are also included, and an accurate model can be constructed representing the reality.

The fourth sub-question was: *How can the prediction model be developed?* The prediction model can be developed by using a CPS. With the CPS, an autonomous architecture is created where the cyber part interacts with the physical part and vice versa. The CPS for the multi-component system consists of a lower-level model and a higher-level model. Within the lower-level model, the root causes of failure are first determined before data collection occurs. It was found that Big Data enables accessible data collection but can take enormous proportions in size. Therefore, a clear vision of what data to collect must be present. After that, data is collected together with feature extraction and dimension reduction. With this data to information part, the health status of each sub-system can be determined and serves as the output of the lower-level model. The higher-level model uses the health status of each sub-system and predicts and visualizes the system's health status using a DBN. At last, decision-making to find the optimal time to maintain the system and the individual sub-systems occurs in the higher-level model.

The fifth sub-question was: *How can the developed prediction model be implemented in the maintenance strategy of the Passenger Boarding Bridge?* The developed prediction model could not be fully implemented in the maintenance strategy of the PBB. In this research, only model verification has been done, including a simulation of the model. The base case was determined based on the sub-system canopy and its components. To account for uncertainties and unclarity, assumptions were made to make the simulation model. The simulation model showed that the developed prediction model functions as it should, regardless of the given input. Therefore, if the correct input data is available, the model can be implemented in the current maintenance strategy.

The last sub-question was: *How does the developed predictive maintenance strategy perform in relation to the turnaround process?* KPIs were determined to see how the developed PdM strategy performs. The KPIs were stated as reliability, the repair time and the repair rate. With the simulation model and synthetic dataset, it can be concluded that the developed PdM strategy can monitor the system under scope and predict its reliability within a certain period. By actively updating, based on new information, the health status of each sub-system at a new time step, proactive maintenance can be executed by choosing the optimal timestep based on the lowest repair rate. With these proactive approaches, failures are prevented, and the MTBF increases, which increases the system's reliability during in-time use and prevents delays in the turnaround process from happening.

Combining the answers to the sub-questions, the research question *How to realize a predictive maintenance strategy for Passenger Boarding Bridges to benefit the airport's turnaround process?* can be answered. It has been concluded that by developing a CPS architecture for the PBB, a predictive maintenance strategy can be realized, which can benefit the airport's turnaround process. By addressing the root causes of the system's failure, adequate data collection can be done, enabling continuous health monitoring of the bridge, its sub-system and its components. With these predictions, decision-making can occur, allowing proactive maintenance moments at which the repair rate is at its lowest while respecting the availability constraints of aircraft stand. With this, the reliability of the PBB is justified and improved, and unwanted downtime during the turnaround process is prevented.

8.2. Limitations and future research

As the usage of actual sensor data was limited due to the unclarity in the data and time limitation, the system was simplified, and a synthetic dataset was used. This limited the research on the following points. First, as the maintenance tickets and especially the maintenance logs could not give clear insight into the root causes of failure for the different sub-systems of the PBB, only research was conducted on the canopy. From here, the validity of the used type of component from the supplier was questioned due to quality differences. Combining this with the data collection of the canopy was also questionable; a simplified model was used for the lower-level model and its resulting output. This limited the model at the cyber level. Therefore, It is recommended that future research focuses on applying multi-model approaches and how to implement this on a multi-component system to reduce the complexity and ease its usage in practice. Second, the higher-level model uses a DBN to indicate the relations and dependencies between the sub-systems and to visualize the system. This research found that the DBN works perfectly; however, determining the conditional probabilities was difficult. This resulted in many assumptions and expert knowledge for this part. The sensitivity analysis shows that the DBN is highly sensitive to the outcome with different probabilities. It is therefore also recommended to explore further if a DBN is the right tool for the cognition layer, and if so, more research must be done for the PBB and the conditional probabilities. At last, within the decision-making model, the input values were determined based on historical data and distributions were used to represent these values. Combining this with the fact that the availability is only based on the time between aircraft handling in the past, it is recommended to get the actual values for the input values of the decision-making model in the configuration layer before implementing it in the real world.

8.3. Recommendations for Schiphol

In this section, the practical recommendations for Schiphol are listed. First, understand which data needs to be collected. As Big Data and sensors can enable data collection of almost everything, this can result in data collected that is not used afterward. Finding the root causes of failure and clearly stating and documenting why and for which purpose, useful data can be collected. With this useful data, information can be extracted to predict upcoming failures. Second, if this root cause analysis results in the failure's cause can not be predicted, this does not mean the search has ended. As PdM is not a step but an enhancement in the maintenance strategy, other ways to prevent upcoming failures must be used. Third, it was found that the maintenance tickets were not functioning as they were supposed to. As the maintenance tickets serve as indicators of the failures happening and maintenance logs were added to indicate what has happened and how it is solved, these tickets must be filled in correctly if needed later on to do a root cause analysis. Fourth, the technical and data-driven parts come together when implementing PdM from theory to practice. As it is easy to focus from a data-driven point of view, it is easy to forget that there are mechanics that are working daily with the PBB. By having close contact with the mechanics, the knowledge needed for the technical part could be more effortless to collect and could even enhance the data-driven part. This is because they are the user of the maintenance strategy and the people who need to understand it. Including them in an earlier phase could make the process more convenient. By acknowledging and overcoming the abovementioned four points, predictive maintenance is possible for the PBB. However, the complexity of the PBB, in combination with suspicion of the high number of failures to predict or overcome, the feasibility of realizing the predictive maintenance for the PBB within the busy world of Schiphol will be challenging.

Bibliography

Achouch, M., Dimitrova, M., Ziane, K., Karganroudi, S. S., Dhouib, R., Ibrahim, H., & Adda, M. (2022). On Predictive Maintenance in Industry 4.0: Overview, Models, and Challenges. *Applied Sciences*.

Airlines for America. (2022). U.S. Passenger Carrier Delay Costs. Retrieved January 18, 2023, from https://www.airlines.org/dataset/u-s-passenger-carrier-delay-costs/

Airport Council International, Air Transport Association, & Airport Coordinators Group. (2020). Worldwide Airport Slot Guidelines.

Akbar, S., Vaimann, T., Asad, B., Kallaste, A., Sardar, M. U., & Kudelina, K. (2023). State-of-the-Art Techniques for Fault Diagnosis in Electrical Machines: Advancements and Future Directions [Number: 17 Publisher: Multidisciplinary Digital Publishing Institute]. *Energies*, *16*(17), 6345. https://doi.org/10.3390/en16176345

Akhbardeh, F., Desell, T., & Zampieri, M. (2020). NLP Tools for Predictive Maintenance Records in MaintNet. *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: System Demonstrations*, 26–32. Retrieved May 1, 2023, from https://aclanthology.org/2020.aacl-demo.5

Atamuradov, V., Medjaher, K., Dersin, P., Lamoureux, B., & Zerhouni, N. (2020). Prognostics and Health Management for Maintenance Practitioners - Review, Implementation and Tools Evaluation. *International Journal of Prognostics and Health Management*, *8*(3). https://doi.org/10.36001/ijphm.2017.v8i3.2667

Aviation Learnings Team. (2020). How Jet Bridge (or Passenger Boarding Bridge) Works? Retrieved February 3, 2023, from https://aviationlearnings.com/passenger-boarding-bridge-jet-bridge-jetty-or-aerobridge-you-name-it/

BayesFusion. (2023). *GeNIe Modeler Programmer's Manual*.

Biard, G., & Nour, G. A. (2021). Industry 4.0 Contribution to Asset Management in the Electrical Industry [Number: 18 Publisher: Multidisciplinary Digital Publishing Institute]. *Sustainability*, *13*(18), 10369. https://doi.org/10.3390/su131810369

Calabrese, F., Regattieri, A., Bortolini, M., Gamberi, M., & Pilati, F. (2021). Predictive Maintenance: A Novel Framework for a Data-Driven, Semi-Supervised, and Partially Online Prognostic Health Management Application in Industries [Number: 8 Publisher: Multidisciplinary Digital Publishing Institute]. *Applied Sciences*, *11*(8), 3380. https://doi.org/10.3390/app11083380

Cannavacciuolo, L., Ferraro, G., Ponsiglione, C., Primario, S., & Quinto, I. (2023). Technological innovation-enabling industry 4.0 paradigm: A systematic literature review. *Technovation*, *124*, 102733. https://doi.org/10.1016/j.technovation.2023.102733

Cao, Q., Zanni-Merk, C., Samet, A., Reich, C., Beuvron, F. d. B. d., Beckmann, A., & Giannetti, C. (2022). KSPMI: A Knowledge-based System for Predictive Maintenance in Industry 4.0. *Robotics and Computer-Integrated Manufacturing*, *74*, 102281. https://doi.org/10.1016/j.rcim.2021.102281

Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., Francisco, R. d. P., Basto, J. P., & Alcalá, S. G. S. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, *137*, 106024. https://doi.org/10.1016/j.cie.2019.106024

Centraal Bureau voor de Statistiek. (2022). Hoeveel passagiers reizen via Nederlandse luchthavens? Retrieved November 30, 2022, from https://www.cbs.nl/nl-nl/visualisaties/verkeer-en-vervoer/personen/passagiers-luchtvaart

DAIR.AI. (2023). Prompt Engineering Guide. Retrieved May 17, 2023, from https://www.promptingguide.ai/

de Alvear Cardenas, J. I., Haegens, T., Hendrix, T., Kerstens, F., Kokee, L., Duim, T., Elsloo, S., & Wildemans, N. (2017). *Aircraft Departure Delay and its Associated Costs for Airlines and Airports in Europe* (tech. rep.). https://doi.org/10.13140/RG.2.2.34307.55847 de Pater, I., & Mitici, M. (2021). Predictive maintenance for multi-component systems of repairables with Remaining-Useful-Life prognostics and a limited stock of spare components. *Reliability Engineering & System Safety*, *214*, 107761. https://doi.org/10.1016/j.ress.2021.107761

Dinh, D.-H., Do, P., & Iung, B. (2022). Multi-level opportunistic predictive maintenance for multi-component systems with economic dependence and assembly/disassembly impacts. *Reliability Engineering & System Safety*, *217*, 108055. https://doi.org/10.1016/j.ress.2021.108055

Fasuludeen, K. F. k., Naveed, N., Anwar, M. N., & Ul, H. M. I. (2021). Production and maintenance in industries: Impact of industry 4.0 [Publisher: Emerald Publishing Limited]. *Industrial Robot: the international journal of robotics research and application*, *49*(3), 461–475. https://doi.org/10.1108/IR-09-2021-0211

Federal Aviation Administration. (2020). Cost of Delay Estimates 2019.

Gashi, M., & Thalmann, S. (2020). Taking Complexity into Account: A Structured Literature Review on Multi-component Systems in the Context of Predictive Maintenance. In M. Themistocleous & M. Papadaki (Eds.), *Information Systems* (pp. 31–44). Springer International Publishing. https://doi.org/10.1007/978-3-030-44322-1_3

Gawde, S., Patil, S., Kumar, S., Kamat, P., Kotecha, K., & Abraham, A. (2023). Multi-fault diagnosis of Industrial Rotating Machines using Data-driven approach : A review of two decades of research. *Engineering Applications of Artificial Intelligence*, *123*, 106139. https://doi.org/10.1016/j.engappai.2023.106139

Gomes, I. P., & Wolf, D. F. (2020). Health Monitoring System for Autonomous Vehicles using Dynamic Bayesian Networks for Diagnosis and Prognosis. *Journal of Intelligent & Robotic Systems*, *101*(1), 19. https://doi.org/10.1007/s10846-020-01293-y

Gupta, V., Mitra, R., Koenig, F., Kumar, M., & Tiwari, M. K. (2023). Predictive maintenance of baggage handling conveyors using IoT. *Computers & Industrial Engineering*, *177*, 109033. https://doi.org/10.1016/j.cie.2023.109033

Hashemian, H. M. (2011). State-of-the-Art Predictive Maintenance Techniques [Conference Name: IEEE Transactions on Instrumentation and Measurement]. *IEEE Transactions on Instrumentation and Measurement*, *60*(1), 226–236. https://doi.org/10.1109/TIM.2010.2047662

Hu, J., Zhang, L., & Liang, W. (2012). Opportunistic predictive maintenance for complex multi-component systems based on DBN-HAZOP model. *Process Safety and Environmental Protection*, *90*(5), 376–388. https://doi.org/10.1016/j.psep.2012.06.004

International Civil Aviation Organization. (2019). Future of Aviation. Retrieved January 18, 2023, from https://www.icao.int/Meetings/FutureOfAviation/Pages/default.aspx

Kamble, S. S., Gunasekaran, A., & Gawankar, S. A. (2018). Sustainable Industry 4.0 framework: A systematic literature review identifying the current trends and future perspectives. *Process Safety and Environmental Protection*, *117*, 408–425. https://doi.org/10.1016/j.psep.2018.05.009

Koenig, F., Found, P. A., Kumar, M., & Rich, N. (2020). Condition-based maintenance for major airport baggage systems [Publisher: Emerald Publishing Limited]. *Journal of Manufacturing Technology Management*, *32*(3), 722–741. https://doi.org/10.1108/JMTM-04-2019-0144

Kwon, D., Hodkiewicz, M. R., Fan, J., Shibutani, T., & Pecht, M. G. (2016). IoT-Based Prognostics and Systems Health Management for Industrial Applications [Conference Name: IEEE Access]. *IEEE Access*, *4*, 3659–3670. https://doi.org/10.1109/ACCESS.2016.2587754

Lee, D., & Pan, R. (2017). Predictive maintenance of complex system with multi-level reliability structure. *International Journal of Production Research*, *55*(16), 4785–4801. https://doi.org/10.1080/00207543.2017.1299947

Lee, J., Bagheri, B., & Kao, H.-A. (2015). A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. *Manufacturing Letters*, *3*, 18–23. https://doi.org/10.1016/j.mfglet.2014.12.001

Lesch, V., Züfle, M., Bauer, A., Iffländer, L., Krupitzer, C., & Kounev, S. (2023). A literature review of IoT and CPS—What they are, and what they are not. *Journal of Systems and Software*, *200*, 111631. https://doi.org/10.1016/j.jss.2023.111631

Lo, L. S. (2023). The CLEAR path: A framework for enhancing information literacy through prompt engineering. *The Journal of Academic Librarianship*, *49*(4), 102720. https://doi.org/10.1016/j.acalib.2023.102720

Martins, F. B., Cardoso, R., Tammela, I., Colombo, D., & Matos, B. A. d. (2018). Applying CBM and PHM concepts with reliability approach for Blowout Preventer (BOP): A literature review [Number:

1]. *Brazilian Journal of Operations & Production Management*, *15*(1), 78–95. https://doi.org/10.14488/BJOPM.2018.v15.n1.a8

Montero Jimenez, J. J., Schwartz, S., Vingerhoeds, R., Grabot, B., & Salaün, M. (2020). Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics. *Journal of Manufacturing Systems*, *56*, 539–557. https://doi.org/10.1016/j.jmsy.2020.07.008

Nemeth, T., Ansari, F., Sihn, W., Haslhofer, B., & Schindler, A. (2018). PriMa-X: A reference model for realizing prescriptive maintenance and assessing its maturity enhanced by machine learning. *Procedia CIRP*, *72*, 1039–1044. https://doi.org/10.1016/j.procir.2018.03.280

Nicolai, R. P., & Dekker, R. (2008). Optimal Maintenance of Multi-component Systems: A Review. In K. A. H. Kobbacy & D. N. P. Murthy (Eds.), *Complex System Maintenance Handbook* (pp. 263–286). Springer. https://doi.org/10.1007/978-1-84800-011-7_11

Niestadt, M. (2021). The future of regional airports: Challenges and opportunities.

Ran, Y., Zhou, X., Lin, P., Wen, Y., & Deng, R. (2019). A Survey of Predictive Maintenance: Systems, Purposes and Approaches [arXiv:1912.07383 [cs, eess]]. Retrieved November 1, 2022, from http://arxiv.org/abs/1912.07383

Rebaiaia, M.-L., & Ait-Kadi, D. (2022). A Remaining Useful Life Model for Optimizing Maintenance cost and Spare-parts replacement of Production Systems in the Context of Sustainability. *IFAC-PapersOnLine*, *55*(10), 1562–1568. https://doi.org/10.1016/j.ifacol.2022.09.613

Sargent, R. (2011). Verification and validation of simulation models. *37*, 166–183. https://doi.org/10.1109/WSC.2010.5679166

Schenkelberg, K., Seidenberg, U., & Ansari, F. (2020). Analyzing the impact of maintenance on profitability using dynamic bayesian networks. *Procedia CIRP*, *88*, 42–47. https://doi.org/10.1016/j.procir.2020.05.008

Schietekat, S., de Waal, A., & Gopaul, K. (2016). Validation & Verification of a Bayesian Network Model for Aircraft Vulnerability.

Schiphol. (2022). Schiphol | Amsterdam Airport Schiphol as our main activity. Retrieved February 24, 2023, from https://www.schiphol.nl/en/schiphol-group/page/amsterdam-airport-schiphol/

Schiphol Group. (2021). Capacity declaration summer season 2022.

Selcuk, S. (2017). Predictive maintenance, its implementation and latest trends [Publisher: IMECHE]. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *231*(9), 1670–1679. https://doi.org/10.1177/0954405415601640

Shafiq, S. I., Sanin, C., & Szczerbicki, E. (2022). Decisional DNA (DDNA) Based Machine Monitoring and Total Productive Maintenance in Industry 4.0 Framework [Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01969722.2021.2018549]. *Cybernetics and Systems*, *53*(5), 510–519. https://doi.org/10.1080/01969722.2021.2018549

Sharp, M., Sexton, T., & Brundage, M. (2017). Toward Semi-Autonomous Information Extraction for Unstructured Maintenance Data in Root Cause Analysis. *513*.

Shin, M., Hwang, S., Kim, B., Seo, S., & Kim, J. (2022). IoT-Based Intelligent Monitoring System Applying RNN [Number: 20 Publisher: Multidisciplinary Digital Publishing Institute]. *Applied Sciences*, *12*(20), 10421. https://doi.org/10.3390/app122010421

Shukla, K., Nefti-Meziani, S., & Davis, S. (2022). A heuristic approach on predictive maintenance techniques: Limitations and scope [Publisher: SAGE Publications]. *Advances in Mechanical Engineering*, *14*(6), 16878132221101009. https://doi.org/10.1177/16878132221101009

Silvestri, L., Forcina, A., Introna, V., Santolamazza, A., & Cesarotti, V. (2020). Maintenance transformation through Industry 4.0 technologies: A systematic literature review. *Computers in Industry*, *123*, 103335. https://doi.org/10.1016/j.compind.2020.103335

Song, L., Wang, L., Wu, J., Liang, J., & Liu, Z. (2021). Integrating Physics and Data Driven Cyber-Physical System for Condition Monitoring of Critical Transmission Components in Smart Production Line [Number: 19 Publisher: Multidisciplinary Digital Publishing Institute]. *Applied Sciences*, *11*(19), 8967. https://doi.org/10.3390/app11198967

Tinga, T., & Loendersloot, R. (2014). Aligning PHM, SHM and CBM by understanding the physical system failure behaviour.

Usuga-Cadavid, J. P., Lamouri, S., Grabot, B., & Fortin, A. (2022). Using deep learning to value free-form text data for predictive maintenance [Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00207543.202

International Journal of Production Research, *60*(14), 4548–4575. https://doi.org/10.1080/00207543.2021.1951868

Van Horenbeek, A., & Pintelon, L. (2013). A dynamic predictive maintenance policy for complex multi-component systems. *Reliability Engineering & System Safety*, *120*, 39–50. https://doi.org/10.1016/j.ress.2013.02.029

van Barneveld, T., & Verheijden, M. (2019). Kans 1 Flexibele onderhoudsplanning op connected VOP's.

van Dinter, R., Tekinerdogan, B., & Catal, C. (2022). Predictive maintenance using digital twins: A systematic literature review. *Information and Software Technology*, *151*, 107008. https://doi.org/10.1016/j.infsof.2022.107008

Verheijden, M. (2018). Roadmap optimaal assetrendement Perceel 2 Vliegtuigafhandeling - Verhogen technische beschikbaarheid.

VolkerWessels. (2021). Maincontract Schiphol en VolkerWessels Infrastructuur verlengd - VolkerWessels. Retrieved December 7, 2022, from https://www.volkerwessels.com/nl/nieuws/maincontract-schiphol-en-volkerwessels-infrastructuur-verlengd

Wang, S., Tomovic, M., & Liu, H. (2016). Chapter 3 - Comprehensive Reliability Design of Aircraft Hydraulic System. In S. Wang, M. Tomovic, & H. Liu (Eds.), *Commercial Aircraft Hydraulic Systems* (pp. 115–169). Academic Press. https://doi.org/10.1016/B978-0-12-419972-9.00003-6

Werbińska-Wojciechowska, S., & Winiarska, K. (2023). Maintenance Performance in the Age of Industry 4.0: A Bibliometric Performance Analysis and a Systematic Literature Review [Number: 3 Publisher: Multidisciplinary Digital Publishing Institute]. *Sensors*, *23*(3), 1409. https://doi.org/10.3390/s23031409

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., & Schmidt, D. C. (2023). A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT [arXiv:2302.11382 [cs]]. Retrieved May 16, 2023, from http://arxiv.org/abs/2302.11382

Zhang, W., Yang, D., & Wang, H. (2019). Data-Driven Methods for Predictive Maintenance of Industrial Equipment: A Survey [Conference Name: IEEE Systems Journal]. *IEEE Systems Journal*, *13*(3), 2213–2227. https://doi.org/10.1109/JSYST.2019.2905565

Zhang, X., Jiang, H., Zheng, B., Li, Z., & Gao, H. (2022). Optimal maintenance period and maintenance sequence planning under imperfect maintenance [_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/qre. *Quality and Reliability Engineering International*, *n/a*(n/a). https://doi.org/10.1002/qre.3192

Zhao, Y., Tong, J., Zhang, L., & Wu, G. (2020). Diagnosis of operational failures and on-demand failures in nuclear power plants: An approach based on dynamic Bayesian networks. *Annals of Nuclear Energy*, *138*, 107181. https://doi.org/10.1016/j.anucene.2019.107181

Zonta, T., da Costa, C. A., da Rosa Righi, R., de Lima, M. J., da Trindade, E. S., & Li, G. P. (2020). Predictive maintenance in the Industry 4.0: A systematic literature review. *Computers & Industrial Engineering*, *150*, 106889. https://doi.org/10.1016/j.cie.2020.106889

A

Scientific research paper

The scientific research paper starts on next page.

Towards a predictive maintenance strategy for Passenger Boarding Bridges at the airport

L. Meijs, Dr. ir. Y. Pang, Prof. dr. R. R. Negenborn, O. Maan, J. van der Lee

Abstract - With Schiphol Airport's flight numbers growing, working assets are essential to ensure on-time processes. The Passenger Boarding Bridge (PBB) is a critical asset in the airport's turnaround process. By ensuring that the asset is working properly, the operational processes can run efficiently. Currently, improving the reliability of the PBB when in use happens after the fault has occurred. With this maintenance strategy, the PBB data is not used to predict the future health state of the PBB. Literature shows that the PBB can be classified as a multi-component system. Research in the predictive maintenance strategy of multi-component systems is still in an early phase. Research until now is more theoretical than practical, and an investigation into applying theoretical knowledge in practice is needed. With the upcoming developments of Industry 4.0, a Cyber-Physical System (CPS) architecture is proposed for a multi-component system. This architecture has been applied to the PBB to develop and use a predictive maintenance strategy for this system. Based on the implementation of a simulation model, the output showed that the proposed CPS architecture enabled the development of a predictive maintenance strategy for the PBB. With this strategy, proactive maintenance is planned while the system's reliability is held on a preset level to ensure a working asset during in-time use.

I. INTRODUCTION

A. Research background

The aviation industry has seen enormous growth (International Civil Aviation Organization, 2019). This growth has led to the will of airports to expand in size and flight numbers (Niestadt, 2021). This increase in capacity needs to be captured between the existing limits of the airport. With airports functioning within the top of their limitations, delays impose a risk for the airports. Delays induce significant costs for the airlines, the primary concern of the airport company (de Alvear Cardenas et al., 2017). It was estimated that the average price of aircraft block time, the time difference an aircraft goes into and out of the blocks, also known as the turnaround time, in the United States was 80.52 dollars per minute delay (Airlines for America, 2022). This resulted in an overall cost of 33 billion dollars due to delays in the United States in 2019 (Federal Aviation Administration, 2020). The PBB is a critical asset in the turnaround process. The bridge ensures that passengers can walk dry and in a comfortable climate regardless of the conditions outside.

B. Problem definition

The growth of the aviation industry is also visible at Schiphol Airport. From 2010 to 2020, Schiphol Airport increased from 45 million to 72 million passengers using the airport. This is an increase of almost 63 percent more passengers using the airport to get from or to their destination (Centraal Bureau voor de Statistiek, 2022). Everything must work correctly to ensure optimal passenger journeys and prevent flight delays. If an aircraft is delayed, passengers can claim compensation for the imposed delay by the airline company. Also, a delay can have an influence on the operational processes. A delay can result in the aircraft being unable to depart within its own time slot. Flight time slots are used as timetables for the airport to regulate all the departing and arriving flights. Within the designated time slot, usually 20 minutes (Schiphol Group, 2021), the airlines can use all airport infrastructure (for example, taxi lanes and runways) necessary for the successful operation of the flight (Airport Council International et al., 2020). However, when an aircraft misses its time slot, it must stay on the ground until a new time slot has been found. Within a busy airport like Schiphol, the flight delay will increase even more.

Through an innovation proposal started by Schiphol, sensors were installed at three PBBs at Schiphol to get more insight into the failure mechanisms of the PBB. Together with the data gathered from the bridge Programmable Logic Controller (PLC), the data is visualized to see if the PBB's failure mechanism could be seen or predicted based on the data. If a prediction could be made about when the PBB will be 'out of order' within a specific time range, the delays jeopardizing the turnaround process would be reduced. Until now, the obtained data has been analyzed on anomalies after failure. This resulted in that only warnings have been implemented, which will become active if certain thresholds are exceeded. Therefore, the potential of using the sensor data in predicting a failure of the PBB before it occurs in the system has not been reached yet. Secondly, the time the operator uses to (de)connect the PBB varies between 40 seconds and more than 5 minutes. Nevertheless, these are short periods of in-use time of the PBB, resulting in a more complex situation than continuous monitoring of an asset used for hours, which is seen more often in literature. This has led to the following problem statement:

Currently, improving the reliability of the Passenger Boarding Bridge when in use, based on the monitored data, takes place after the fault has occurred. Which, therefore, does not use the data's possibilities to predict the future health state of the Passenger Boarding Bridge to its maximum.

C. Research objective

The research goal has been formulated based on the problem definition described above. The research goal is to develop a prediction model to forecast an impending failure of the PBB to prevent downtime of the PBB during in-time use. This forecast must then result in maintenance activities of the PBB being done proactively, or a real-time gate switching could be suggested. All this together must lead to a decrease in the time delay of the turnaround process of the aircraft, which is currently directly affected if failure of the PBB occurs.

Combining the problem definition and research objective, the following research question is formulated and answered in this paper: *How to realize a predictive maintenance strategy for Passenger Boarding Bridges to benefit the airport's turnaround process?*

D. Scope

This research's scope is limited to the 65 CMIC Tianda PBBs at Schiphol Airport, their maintenance tickets and related failures. These maintenance tickets are from 1 April 2019 to 31 December 2022. The sensor data from the sensors installed by the main contractor, VolkerInfra, are from three CMIC Tianda bridges, located at stands D16, D18 and D51. This sensor data is gathered starting from 1 January 2020. Outside the scope of this research are the maintenance tickets and sensor data related to the bumper due to an already investigated innovation proposal, real-time switching of aircraft stands, the maintenance tickets related to human errors and reducing the planned maintenance moments. The latter is because this research will look into decreasing the unplanned maintenance moments.

E. Methodology

The following methodology has been executed to answer the research question. First, the current situation of the PBB at Schiphol was investigated. In this analysis, the working principle of the PBB is presented to know how the bridge is operated. From maintenance logs provided by Volkerinfra, qualitative data is analyzed to find the root causes of failure. In the analysis, the maintenance logs are combined with the Failure Mode, Effect and Critical Analysis (FMECA) and the Site Acceptance Testing (SAT) protocol of the PBB to link the failure mechanism to the sub-systems of the bridge. From here, the bridge's critical sub-systems regarding the operation of the PBB were defined. Next, literature research was executed to investigate what has already been written about predictive maintenance. This will provide a clear overview of predictive maintenance and the knowledge needed about state-of-the-art techniques for predictive maintenance. A CPS architecture was used in this research (Lee et al., 2015; Song et al., 2021). After the model was developed, verification of the model was done. The model was simulated to see the performance relative to the current maintenance strategy of the PBB.

II. RESEARCH OBJECT

A. Passenger Boarding Bridge

Figure 1 shows the research object in this paper, the PBB. The PBB can be divided into seven subcomponents: the telescopic tunnels, the elevation system, the wheel bogies, the cabin, the canopy, the trim arm and the roller door. The PBB is used in two situations. The first situation is the connection towards an arriving aircraft. Here, the PBB is manually operated towards the aircraft. The second situation is decoupling from the aircraft. Although this is started manually by pressing the parking button, the parking procedure is done automatically towards a predefined parking position.

FIG. 1. The Passenger Boarding Bridge

B. Current maintenance situation

At Schiphol Airport, two maintenance strategies are already used to maintain the PBB. The first is reactive maintenance: if there is a problem with the PBB, a maintenance mechanic of VolkerInfra must come to inspect the situation. A maintenance ticket needs to be made and sent to the assigned mechanic. The maintenance ticket can originate from three sources: from the bridge's operator, business operations, or directly from the bridge due to the current still-in-progress Smart Ticketing innovation. When the maintenance mechanic gets the order to inspect the PBB, they must be at the PBB within 15 minutes if the handling of the aircraft encounters a disturbance. Otherwise, this time to arrival is 60 minutes. The second maintenance strategy is the preventive maintenance strategy. These preventive inspection intervals are determined by using the FMECA of the PBB. To get more insight into the behavior of the PBB, VolkerInfra installed extra sensors to monitor the bridge's condition in real-time. However, no arguments were documented as to why the specific additional sensors were installed and why other data from the bridge were excluded from the monitoring. Therefore, the next section will analyze the failure mechanisms of the PBB and relate it to the different sub-systems to, in the end, evaluate if the current data is sufficient for a prediction model or if extra data needs to be collected.

C. Failure mechanisms

Failure of the bridge is defined as a problem of the bridge that results in a stoppage of aircraft handling. Based on the classifications within the maintenance tickets, the following was concluded. First, when viewing the causes of the different sub-system failure tickets, in all cases, more than 59 percent are classified as technical causes compared to 2 percent for wear. A conclusion could be drawn that degradation of the system is not a primary cause of failure; therefore, it could be suggested that prediction models based on life cycles are not of interest to this research. However, this conclusion could be wrong. It can be said that failure due to degradation is prevented due to the preventive maintenance strategy. Nevertheless, the argument that the PBB has a short use interval and, therefore, degradation of the PBB is not going rapidly can also be used. For each sub-system, the reason for the maintenance ticket was clustered by the fault tree of VolkerInfra. However, most of these classifications were too general to derive conclusions from to know what caused the failure. Without the root cause of failure, an accurate prediction model can not be made to predict and prevent the impending failure. It was concluded that no evaluation of the monitored data based on the placed sensors could be done due to the unknown root causes of failure.

III. PREDICTIVE MAINTENANCE

Predictive maintenance (PdM) relies on the continuous monitoring of the actual status of the system. Based on the data collected, a prediction will be made of the remaining time the component or the system will be in function or is likely to fail. Then, a trade-off will be made between the condition of the component and the maintenance frequency (Ran et al., 2019; Zhang et al., 2019). This proactive method will thus, instead of time-based maintenance, only plan maintenance when needed and reduce unnecessary downtime due to on-time failure detection (Carvalho et al., 2019; van Dinter et al., 2022). To achieve a PdM strategy, the following steps need to be followed (Achouch et al., 2022; Selcuk, 2017). First, the researched object needs to be understood. This means knowing how the system operates, why the object fails, what is already measured, and what the goal of the strategy is. For effectively applying the strategy, the root cause of failure must be known instead of only addressing the symptoms of failure. Second, data must be gathered, understood, and prepared for further use. Here, understanding the object and knowing the root cause play a role, as gathering data about conditions that are not used in the end should be avoided. Third, the data must be used in a model for predicting the upcoming failure or the future health state of the object. Fourth, the model's accuracy must be known to assess how the model will perform. After that, the model should be deployed in the researched situation. If the model is not first evaluated correctly, it will fail to describe the real world properly. In this step, it also means that the steps explained above are executed properly and/or addressed otherwise in the evaluation before deploying the model. The fifth, also the last step, is decision-making. Based on the PdM goal, decision-making takes place based on the outcome of the used model.

Within the literature, three approaches for a prediction model can be classified: physical-based, knowledge-based and data-driven. A combination of the three approaches can also be used, also known as a multi-model approach. For a complex system, these single-model approaches only partly address the diagnostic and prognostic task of the system. A multi-model approach could be implemented to overcome the complexity of the system. In figure 2, combinations of the approaches can be seen. Besides the combinations that can be made between the different approaches, the architecture of the multi-model approach is important. The models can be in series, parallel to each other, or a model can be embedded in another model. (Montero Jimenez et al., 2020).

IV. CYBER-PHYSICAL SYSTEM ARCHITECTURE

As presented in Lee et al. (2015), the CPS architecture is proposed to develop a predictive maintenance strategy. The choice for a CPS architecture was based on the fact that the system can be autonomous by integrating computing, monitoring and control of the physical components. By enabling the system to affect the health status of the components in the cyber part and to be able to do maintenance activities for the physical part. The system enables itself to control its health.

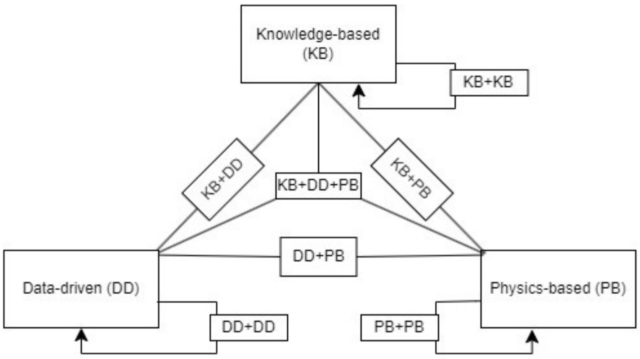


FIG. 2. The possible combinations of single-model approaches, recreated from Montero Jimenez et al. (2020)

The first layer of the CPS architecture is the connection layer. This layer collects data from different sources to connect the physical object with the virtual world. The first step in the connection layer consists of using the information from the analyzed maintenance tickets and experts' knowledge. With this information, the root causes of failure can be determined. Then, the second step in the connection layer is the data extraction from different sources.

The data collected in the first layer are the raw signals directly from the system. These raw signals are collected at a high frequency and are large in amount. Feature extraction and dimension reduction occur in the conversion layer to ensure that the collected data becomes valuable information.

Each sub-system's health status prediction individually takes place in the cyber layer. In this stage, no dependencies or correlations between different sub-systems will influence the investigated sub-system's predicted health status. There can be many factors for the sub-systems as to why the component fails. Therefore, the sub-systems were analyzed in the first layer. Based on the found root causes, a multi-model approach is suggested as the amount of data and models could increase with the increment of the failure models. However, this depends on the specific sub-system and failure modes. This is the first part based on the acquired data collected. The second part within the multi-model approach for the sub-system's health assessment is a data-driven model to assess the degradation of the sub-system based on expert knowledge or a degradation model from the literature. Within the cyber layer, the prediction model for assessing the sub-system's health will be trained by historical data to understand the behavior of the bridge. This results in an offline and online phase within the cyber layer.

At the cognition level, the system as a whole will be evaluated to determine the health status. A Bayesian Network (BN) is used. The variables represented by the nodes in the BN can be categorized into four groups: component nodes, sub-system nodes, failure mechanism nodes and system nodes. The failure mechanism nodes contain variables not linked to a specific component based on the maintenance ticket analysis or expert knowledge. This means that this node directly influences the system's health status. Next, a Dynamic Bayesian Network (DBN) is created. A general indication of the timeslices is visualized in figure 3. The approach for the DBN is based on the model of Gomes and Wolf (2020).

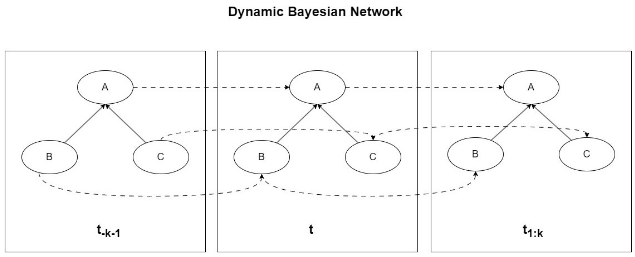


FIG. 3. Proposed three timeframes DBN model, based on Gomes and Wolf (2020)

At the top level of the proposed CPS architecture, decisions are made for maintenance activities. The decision-making model is based on the economic grouping model presented in the paper of Van Horenbeek and Pintelon (2013). However, in this research, the model is reformed with the availability and reliability of the system as objectives. The expected repair cost rate will be replaced by the expected repair rate (RR), equation 4.1. The expected repair rate is related to the time it costs to repair the specific sub-system during maintenance. The expected repair rate can be distinguished into two elements: the corrective repair time and the proactive repair time. It is assumed that the proactive repair time is less than the corrective repair time because when a sudden failure of the sub-system happens, an investigation needs to be done into what caused the failure. The expected repair rate is determined as the sum of the expected corrective and proactive repair cost divided by the time spent since the last maintenance moment Δt, where $\Delta t = t - t_0$. F_i the failure probability distribution of sub-system i. It represents the cumulative distribution function of the random variable "time to failure". The expected setup cost rate is replaced by the expected arrival rate (AR). The expected arrival rate is related to the time it costs for the maintenance mechanism to arrive at the PBB, specified as AT, divided by the time spent since the last maintenance moment. The expected loss of productivity is replaced with the time the system is out of order due to corrective repair or proactive repair, which is the same as the expected repair rate. Therefore, the time of productivity loss is not considered separate but included in the repair time. Second, the maintenance slots and resource availability must be considered. For resource availability, the works schedule of the maintenance mechanics ($m(t)$) can be incorporated into the model, equations 4.3 and 4.4. The different factors for speeding up the process with multiple mechanics are alpha for the site and beta for the shop. A distinction is made between repairs at the site (CRT_i^{site} and $m(t)$) and the time related to the sub-system repair time at the workshop (CRT_i^{shop} and PRT_i^{shop}).

$$RR_i(t) = \frac{CRT_i * F_i(t) + PRT_i(t) * (1 - F_i(t))}{\Delta t} \quad (4.1)$$

$$AR_i(t) = \frac{AT_i}{\Delta t} \quad (4.2)$$

$$CRT_i(t) = \frac{CRT_i^{site}}{\alpha * m(t)} + \frac{CRT_i^{shop}}{\beta * m(t)} \quad (4.3)$$

$$PRT_i(t) = \frac{PRT_i^{site}}{\alpha * m(t)} + \frac{PRT_i^{shop}}{\beta * m(t)} \quad (4.4)$$

Combining the expected repair rate with the expected arrival rate, the expected total out-of-order rate for sub-system i is represented by:

$$TR_i(t) = RR_i(t) + AR_i(t) \quad (4.5)$$

By minimizing the total time it takes to do the maintenance activities, the local optimal PdM moment can be determined.

The global strategy looks at the system as a whole. By using the opportunity to do maintenance not at one sub-system but at multiple sub-systems during one planned maintenance activity, the out-of-order moments of the system are reduced. This results in maintenance groups being formed. The maintenance group can consist of k sub-systems to do maintenance at one moment, with the sub-systems outside the group indicated as NC. First, the expected repair rate of the group, equation 4.6, is calculated. This will be done by summing up the corrective and proactive repair times of the sub-systems k in optimal group G. A reduction coefficient omega is used. With this omega, the influence of grouping sub-systems where, due to structural dependencies, the repair time could be reduced is taken into account. Second, the expected arrival rate of the group is calculated in equation 4.7. Here, the arrival time for doing maintenance activities for group G will be divided by the time spent since the last maintenance moment. The same as in the local strategy, the expected total rate of the group is the expected repair rate of the group together with the expected arrival rate of the group. By minimizing the expected total rate of the group, equation 4.8, the optimal time to do maintenance for the group can be found. The procedure for finding the system optimum and global PdM strategy follows the same approach as presented in the paper of Van Horenbeek and Pintelon (2013); if the expected total maintenance rate of the system, equation 4.9, based on the new maintenance group is less than the expected total maintenance rate based on adding the total rates of the individual sub-systems, the next sub-system is added to the group, and a new total rate is calculated. This procedure will be repeated until the newly formed group has a higher total maintenance repair rate of the system than with the currently determined group. Then, the current group will be the optimal group for combining maintenance activities, with the other sub-systems being repaired independently.

$$RR_{group}(t) = \frac{\sum_{i=1}^{k}(\omega_{CRT} CRT_i(t) \cdot F_i(t) + \omega_{PRT} PRT_i(t) \cdot (1 - F_i(t)))}{\Delta t} \quad (4.6)$$

$$AR_{group}(t) = \frac{AT_{group}}{\Delta t} \quad (4.7)$$

$$TR_{group}(t) = RR_{group}(t) + AR_{group}(t) \quad (4.8)$$

$$TR_{sys} = TR_{group}^{opt} + \sum_{i=k+1}^{NC} TR_i^{opt} \quad (4.9)$$

V. SIMULATION MODEL

A. Bounderies & Assumptions

Due to the complexity of this multi-component system, it is impossible to implement the whole PBB in the proposed architecture due to time constraints. Therefore, only the canopy will be analyzed to prove the proposed CPS architecture.

B. Objective & KPI's

The CPS architecture must provide the following output to show the model's impact. In the lower-level model, the health status of the different components is determined based on various failure mechanisms. Here, the output of the lower-level model will be the health status of the components. To show the impact of the model and compare it to the current situation, the reliability of the components will be used. The reliability is stated as $R(t) = 1 - F(t)$. F(t) is the cumulative probability of the failure, the probability the component or system will fail over time. The exponential distribution will be used with one divided by the MTBF as lambda. With the health status monitored and proactive maintenance being done, the MTBF is expected to be enlarged. This then results in increased reliability.

The second KPI is the maintenance repair time. Schiphol wants to have the availability of the aircraft stand as well as the PBB as high as possible. An inoperable PBB is unwanted, and fast repairments need to take place to have a functional PBB when the turnaround process starts. With the assumption that proactive repairments will be faster than corrective repairments, this KPI indicates that the model's output should be a planning of proactive maintenance moments.

The higher-level model should plan maintenance moments when the aircraft stand is not in use to reduce unexpected failures during in-time use. However, the availability of the aircraft stand is wanted to be as high as possible. Therefore, the third KPI is the repair rate. This means that the repair time over bridge uses is wanted to be as low as possible to execute maintenance activities.

C. Model implementation

With the canopy only in scope, the system role will shift from the PBB to the canopy. First, a base case is defined. Within the canopy, two segments are responsible for being able to extend or retract. One segment is for the left side of the canopy, and one is for the right side. The segments can be divided into the drive, the limit switches and the mechanical part. From the maintenance tickets provided by VolkerInfra, the MTBF, the reliability and corrective repair were determined. Second, the CPS architecture was implemented on the canopy. As for the canopy, the causes of failure are not clear yet. First, the preliminary step in the connection layer will start with maintenance log analysis and gaining knowledge from experts. It was concluded that maintenance log analysis could not provide the root causes of failure for well-considered data collection; expert knowledge was consulted. For the canopy drive, it became clear that not the drive as a whole but the torque limiter within the drive was the cause of failure. To determine the health of the canopy, the drive can be monitored in three non-destructive ways: temperature, vibration and motor current (Hashemian, 2011). Newly installed current sensors were installed to collect data for the connection layer. In the conversion layer, this data was converted to information. However, due to questions of data validity and time constraints, no further research was done on the current data and assumptions were made to use a simplified lower model output. To prove the architecture, the following assumptions were made. First, to prove the CPS architecture, the condition of the drive can be monitored by measuring the temperature, current and vibration of the drive. Second, as CPS architecture is verified by using a simplified model, the sensors that provide information about the condition of the gas spring and the limit switch will be out of scope as this will be modeled in the same way as the drive health indicators. Third, an alternative approach is used due to the lack of data to directly estimate the drive's health status based on the sensors. The sensors are considered nodes within a BN to demonstrate their influence on the overall health state of the drive.

Using a simplified lower-level input, the higher-level model can be modeled. The higher-level model consists of building and verifying the cognition layer and the configuration layer. For the DBN, first, the BN needs to be specified. The procedure in Schietekat et al. (2016) is used to build the BN in the software package GeNIe 4.0 Academic (BayesFusion, 2023), and its structure, variables and their states are visualized in figure 4. The parameters' prior probabilities are determined based on expert knowledge. For the inference technique used in this model, the health status over time, based on evidence, is needed. This means that forward inference is used. Afterward, the BN network was verified and used as input for the DBN. The input for the DBN is the BN combined with the transition of the states over time from the components and canopy. For the time step, one aircraft handling is assumed to happen at a time step. This means that the DBN predicts the health of the canopy for upcoming aircraft handling.

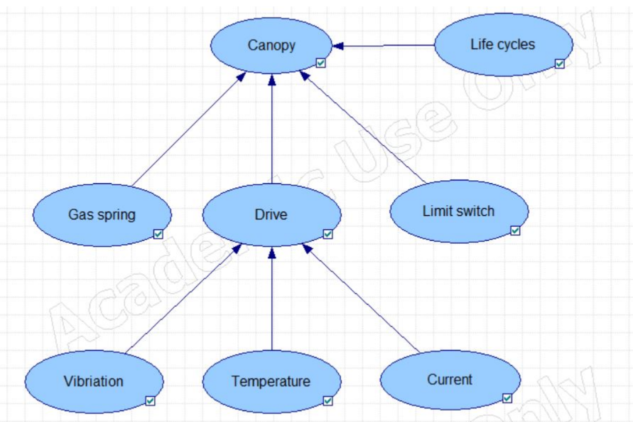


FIG. 4. The Bayesian Network

The output of the cognition level serves as input for the configuration layer. Here, decision-making will be based on the formulas presented in the previous chapter.

VI. RESULTS & DISCUSSION

The base case input was used to determine the optimal maintenance moment for the components individually. After that, the global opportunistic model was used to determine the optimal maintenance group together with the group's optimal maintenance moment. It was shown that the model can determine the optimal maintenance moment for the sub-systems individually and for the optimal maintenance group by finding the lowest repair rate while respecting the availability constraints. A synthetic data set was used for the sensor readings of the first 100 aircraft handlings. Implementing the synthetic dataset in the cognition layer, the DBN updates the failure probability of the drive, and a new decision-making of the optimal maintenance moment is made. The results show that the higher-level model can update the model's reliability and choose the maintenance moment with the lowest repair rate while respecting the availability constraint. When applying the decision-making model to the synthetic dataset, the result shows that the optimal maintenance moment is shifted to a later bridge use. Due to the reliability threshold being otherwise exceeded, the drive needs maintenance within 12 bridge uses. In this case, the availability allows maintenance, but this is not always true. Also, in this model, it is assumed that maintenance can be planned and executed immediately. In reality, this does not have to be the case. It was seen that the procedure to find the optimal maintenance group by combining the first and second sub-systems placed for maintenance was, in this case, not beneficial for the system as the time between them was too long. As the optimal time to do maintenance of the limit switch and gas spring was only 13 bridge uses away, it was seen that if these two sub-systems were combined in a maintenance group, the expected total repair rate of the system would be lower than maintaining these sub-systems individually. By determining this optimal maintenance moment, the proactive repair time can be used, which was assumed less than the corrective repair time. This proactive repair time is also investigated in terms of how it influences the optimal maintenance moment and repair rate for the base and synthetic dataset case. The results showed that the height of the proactive repair time can influence the outcome of the optimal time to do maintenance for the gas spring if a proactive repair time of 0.7 or higher times the corrective repair time is used. For the global opportunistic model, the height of the proactive repair time only influenced the outcome for a proactive repair time as the order of maintenance is shifted due to the earlier optimal time to do maintenance for the gas spring. However, it did not change that it was not beneficial to use the global opportunistic strategy to form a maintenance group of the first and second sub-systems that require maintenance. The influence of the number of mechanics on the repair rate was significant; it was seen that increasing the mechanics from only one to two already decreased the total repair rate from 5 to 14 percent for the local strategy of the gas spring and 25 percent for the global strategy in the synthetic case. However, it did not change the optimal maintenance moment.

VII. CONCLUSION & RECOMMENDATIONS

A. Conclusion

In this paper, an understandable architecture for a predictive maintenance strategy for a multi-component system at the airport is proposed. It has been found that the development of a predictive maintenance strategy for multi-component systems is still in an early phase; nevertheless, with the rise of Industry 4.0, the developments captured within will enhance the research of predictive maintenance for multi-component systems. The use of data-driven methods in combination with big data enables the monitoring of complex systems. Using a multi-model approach, this increasing complexity, as the system will have more components, can be countered and made accessible and understandable. This led to the development of a CPS architecture for the PBB. It is concluded that by implementing this architecture, a predictive maintenance strategy can be developed, which can benefit the airport's turnaround process. By addressing the root causes of the system's failure, adequate data collection can be done, enabling continuous health monitoring of the bridge, its sub-system and its components. With these predictions, decision-making can occur, allowing proactive maintenance moments at which the repair rate is at its lowest while respecting the availability constraints of aircraft stand. With this, the reliability of the PBB is justified and improved, and unwanted downtime during the turnaround process is prevented.

B. Recommendations for further research

The simulation model used was not the preferred choice initially in this research. Due to questions about the validity of the used data and assessed components, a simplified model was used to show the impact of using a theoretical architecture in practice. It is therefore recommended that future research focuses on applying multi-model approaches and how to implement this on a multi-component system to reduce the complexity and ease its usage in practice. This research found that the DBN works perfectly; however, determining the conditional probabilities was difficult. This resulted in many assumptions and expert knowledge for this part. The sensitivity analysis shows that the DBN is highly sensitive to the outcome with different probabilities. It is therefore also recommended to explore further if a DBN is the right tool for the cognition layer, and if so, more research must be done for the PBB and the conditional probabilities. At last, within the decision-making model, the input values were determined based on historical data and distributions were used to represent these values. Combining this with the fact that the availability is only based on the time between aircraft handling in the past, it is recommended to get the actual values for the input values of the decision-making model in the configuration layer before implementing it in the real world.

REFERENCES

Achouch, M., Dimitrova, M., Ziane, K., Karganroudi, S. S., Dhouib, R., Ibrahim, H., & Adda, M. (2022). On Predictive Maintenance in Industry 4.0: Overview, Models, and Challenges. *Applied Sciences*.

Airlines for America. (2022, July). U.S. Passenger Carrier Delay Costs. Retrieved January 18, 2023, from https://www.airlines.org/dataset/u-s-passenger-carrier-delay-costs/

Airport Council International, Air Transport Association, & Airport Coordinators Group. (2020, June). Worldwide Airport Slot Guidelines.

BayesFusion. (2023, September). *GeNIe Modeler Programmer's Manual*.

Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., Francisco, R. d. P., Basto, J. P., & Alcalá, S. G. S. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, *137*, 106024. https://doi.org/10.1016/j.cie.2019.106024

Centraal Bureau voor de Statistiek. (2022). Hoeveel passagiers reizen via Nederlandse luchthavens? Retrieved November 30, 2022, from https://www.cbs.nl/nl-nl/visualisaties/verkeer-en-vervoer/personen/passagiers-luchtvaart

de Alvear Cardenas, J. I., Haegens, T., Hendrix, T., Kerstens, F., Kokee, L., Duim, T., Elsloo, S., & Wildemans, N. (2017, June). *Aircraft Departure Delay and its Associated Costs for Airlines and Airports in Europe* (tech. rep.). https://doi.org/10.13140/RG.2.2.34307.55847

Federal Aviation Administration. (2020). Cost of Delay Estimates 2019.

Gomes, I. P., & Wolf, D. F. (2020). Health Monitoring System for Autonomous Vehicles using Dynamic Bayesian Networks for Diagnosis and Prognosis. *Journal of Intelligent & Robotic Systems*, *101*(1), 19. https://doi.org/10.1007/s10846-020-01293-y

Hashemian, H. M. (2011). State-of-the-Art Predictive Maintenance Techniques [Conference Name: IEEE Transactions on Instrumentation and Measurement]. *IEEE Transactions on Instrumentation and Measurement*, *60*(1), 226–236. https://doi.org/10.1109/TIM.2010.2047662

International Civil Aviation Organization. (2019). Future of Aviation. Retrieved January 18, 2023, from https://www.icao.int/Meetings/FutureOfAviation/Pages/default.aspx

Lee, J., Bagheri, B., & Kao, H.-A. (2015). A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. *Manufacturing Letters*, *3*, 18–23. https://doi.org/10.1016/j.mfglet.2014.12.001

Montero Jimenez, J. J., Schwartz, S., Vingerhoeds, R., Grabot, B., & Salaün, M. (2020). Towards multimodel approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics. *Journal of Manufacturing Systems*, *56*, 539–557. https://doi.org/10.1016/j.jmsy.2020.07.008

Niestadt, M. (2021). The future of regional airports: Challenges and opportunities.

Ran, Y., Zhou, X., Lin, P., Wen, Y., & Deng, R. (2019, December). A Survey of Predictive Maintenance: Systems, Purposes and Approaches [arXiv:1912.07383 [cs, eess]]. Retrieved November 1, 2022, from http://arxiv.org/abs/1912.07383

Schietekat, S., de Waal, A., & Gopaul, K. (2016). Validation & Verification of a Bayesian Network Model for Aircraft Vulnerability.

Schiphol Group. (2021, September). Capacity declaration summer season 2022.

Selcuk, S. (2017). Predictive maintenance, its implementation and latest trends [Publisher: IMECHE]. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *231*(9), 1670–1679. https://doi.org/10.1177/0954405415601640

Song, L., Wang, L., Wu, J., Liang, J., & Liu, Z. (2021). Integrating Physics and Data Driven Cyber-Physical System for Condition Monitoring of Critical Transmission Components in Smart Production Line [Number: 19 Publisher: Multidisciplinary Digital Publishing Institute]. *Applied Sciences*, *11*(19), 8967. https://doi.org/10.3390/app11198967

Van Horenbeek, A., & Pintelon, L. (2013). A dynamic predictive maintenance policy for complex multicomponent systems. *Reliability Engineering & System Safety*, *120*, 39–50. https://doi.org/10.1016/j.ress.2013.02.029

van Dinter, R., Tekinerdogan, B., & Catal, C. (2022). Predictive maintenance using digital twins: A systematic literature review. *Information and Software Technology*, *151*, 107008. https://doi.org/10.1016/j.infsof.2022.107008

Zhang, W., Yang, D., & Wang, H. (2019). Data-Driven Methods for Predictive Maintenance of Industrial Equipment: A Survey [Conference Name: IEEE Systems Journal]. *IEEE Systems Journal*, *13*(3), 2213–2227. https://doi.org/10.1109/JSYST.2019.2905565

B

PLC signals

PLC address	PLC tag / beschrijving	Type	Kanaal
I0.0	Auto_lvl_Raise_Sig	0/1	055-001
I0.1	I_Manual	0/1	055-002
I0.2	Auto_lvl_Desc_Sig	0/1	055-003
I0.3	I_Auto	0/1	055-004
I0.4	Auto_lvl_up_Lim	0/1	055-005
I0.5	Joystick_Forw	0/1	055-006
I0.6	Auto_lvl_low_lim	0/1	055-007
I0.7	Joystick_Backw	0/1	055-008
I1.0	Joystick_left_Rot	0/1	055-009
I1.1	Cabin_left_Rot	0/1	055-010
I1.2	joystick_right_Rot	0/1	055-011
I1.3	Cabin_right_Rot	0/1	055-012
I1.4	Tunnel_Lift	0/1	055-013
I1.5	Canopy_Extend	0/1	055-014
I1.6	Tunnel_Descent	0/1	055-015
I1.7	Canopy_Retract	0/1	055-016
I2.0	Environment_Temp	0/1	055-017
I2.1	Joystick_Enable	0/1	055-018
I2.2	Cabin_Floor_Heater	0/1	055-019
I2.3	Emergency_Stop	0/1	055-020
I2.4	Glass_Heater	0/1	055-021
I2.5	Stop_prox_switch_L	0/1	055-022
I2.6	Claxon_Clax	0/1	055-023
I2.7	Stop_prox_switch_M	0/1	055-024
I3.0	Stop_prox_switch_R	0/1	055-025
I3.1	Cab_Rot_lim_R	0/1	055-026
I3.2	Ultra_spd_Red_L	0/1	055-027
I3.3	Left_Canopy_ext_Lim	0/1	055-028
I3.4	Ultra_spd_Red_R	0/1	055-029
I3.5	Left_Canopy_Pos_1	0/1	055-030
I3.6	Cab_Rot_lim_L	0/1	055-031
I3.7	Left_Canopy_Pos_2	0/1	055-032
I4.0	Left_Canopy_retr_Lim	0/1	055-033
I4.1	Right_Canopy_retr_Lim	0/1	055-034
I4.2	Right_Canopy_ext_Lim	0/1	055-035
I4.3	Cabin_left_rot_Stop	0/1	055-036
I4.4	Right_Canopy_Pos_1	0/1	055-037
I4.5	Cabin_right_rot_Stop	0/1	055-038
I4.6	Right_Canopy_Pos_2	0/1	055-039
I4.7	Roll_door_Open	0/1	055-040
I5.0	Roll_door_close	0/1	055-041
I5.1	High_limit	0/1	055-042
I5.2	A_L_Extend_limit	0/1	055-043
I5.3	Auto_leveling_arm_touch_airplane	0/1	055-044
I5.4	A_L_Retract_limit	0/1	055-045
I5.5	Roll_door_window_protection	0/1	055-046
I5.6	Low_Limit	0/1	055-047
I5.7	Autoleveling_motor_trip	0/1	055-048
I6.0	Point_to_go	0/1	055-049
I6.1	Sec_lim_Switch_R	0/1	055-050
I6.2	Auto_lvl_arm_Gate	0/1	055-051
I6.3	Left_Canopy_power_trip	0/1	055-052
I6.4	Sec_lim_Switch_L	0/1	055-053
I6.5	Right_Canopy_power_trip	0/1	055-054
I6.6	Sec_lim_Switch_M	0/1	055-055
I6.7	Auto_Parking	0/1	055-056
I7.0	Lamp_Test	0/1	055-057
I7.1	Roller_door_trip	0/1	055-058
I7.2	Cabin_rot_motor_trip	0/1	055-059
I7.3	Roller_door_open	0/1	055-060
I7.4	Joystick_power_fail	0/1	055-061
I7.5	Roller_door_close	0/1	055-062
I7.6	Floor_heater_trip	0/1	055-063
I7.7	Window_heater_trip	0/1	055-064
I8.0	Cabin_motor_ISO	0/1	055-065
I8.2	Left_Canopy_ISO	0/1	055-067
I8.4	Right_Canopy_ISO	0/1	055-069

Figure B.1: The first part of signals coming from the PBB PLC

PLC address	PLC tag / beschrijving	Type	Kanaal
I8.6	Roll_door_ISO	0/1	055-071
I10.0	EMS_Relay_Active	0/1	055-081
I10.1	Alarm_power_trip	0/1	055-082
I10.2	Main_power_Ctrl	0/1	055-083
I10.3	Ctrl_power_trip	0/1	055-084
I10.4	Valve_power_trip	0/1	055-085
I10.6	EMS_power_trip	0/1	055-087
I10.7	Reset	0/1	055-088
I11.0	Password_cancel	0/1	055-089
I11.1	Bypass	0/1	055-090
I11.2	Auto_lvl_arm_extend	0/1	055-091
I11.3	PBB_invalid	0/1	055-092
I11.4	Auto_lvl_arm_retract	0/1	055-093
I11.5	Maintanance	0/1	055-094
I11.6	Travel_motor_brake_trip	0/1	055-095
I11.7	Arrester	0/1	055-096
I12.0	Converter1_fault	0/1	055-097
I12.1	UPS_batt_low	0/1	055-098
I12.2	Main_power_fault	0/1	055-099
I12.3	Glass_window_heater_trip	0/1	055-100
I12.4	UPS_batt_fault	0/1	055-101
I12.6	UPS_batt_supply	0/1	055-103
I14.0	Extend_slow	0/1	055-113
I14.1	Ultimate_Extend_limit	0/1	055-114
I14.2	Retract_slow	0/1	055-115
I14.3	Ultimate_retract_limit	0/1	055-116
I14.4	Full_extend_stop	0/1	055-117
I14.6	Full_retract_stop	0/1	055-119
I15.0	Rotunda_left_lim	0/1	055-121
I15.1	Up_slope_lim	0/1	055-122
I15.2	Rotunda_right_lim	0/1	055-123
I15.3	down_slope_lim	0/1	055-124
I15.4	Rotunda_left_stop	0/1	055-125
I15.5	Tunnel_light_push_but	0/1	055-126
I15.6	Rotunda_right_sto	0/1	055-127
I17.0	EMS	0/1	055-137
I17.1	Ultimate_steer_lim_R	0/1	055-138
I17.2	Steer_left_Stop	0/1	055-139
I17.3	Pump_ISO	0/1	055-140
I17.4	Steer_right_stop	0/1	055-141
I17.5	Pump_motor_trip	0/1	055-142
I17.6	Ultimate_steer_lim_L	0/1	055-143
I17.7	Oil_Temp_High	0/1	055-144
I18.0	Oil_Press_High	0/1	055-145
I18.1	VDGS_run	0/1	055-146
I18.2	Inlet_Oil_filter_fault	0/1	055-147
I18.3	VDGS_prohibit	0/1	055-148
I18.4	Oil_lvl_Low	0/1	055-149
I18.5	Bogie_string_switch	0/1	055-150
I18.6	Outlet_Oil_filter_fault	0/1	055-151
I18.7	Left_wheel_ISO	0/1	055-152
I19.0	Right_wheel_ISO	0/1	055-153
I19.1	Maint_available	0/1	055-154
I19.2	Maint_connection	0/1	055-155
I19.3	Maint_lift	0/1	055-156
I19.4	Maint_forward	0/1	055-157
I19.5	Maint_Descent	0/1	055-158
I19.6	Maint_Backward	0/1	055-159
I19.7	Maint_bogie_left_rot	0/1	055-160
I20.0	Maint_bogie_right_rot	0/1	055-161
I20.2	Maint_cabin_left_rot	0/1	055-163
I20.4	Maint_cabin_right_rot	0/1	055-165
Q0.0	Heater_Floor_enable	0/1	055-169
Q0.1	Q_Manual	0/1	055-170
Q0.2	Heater_Glass_enable	0/1	055-171
Q0.3	Q_Auto	0/1	055-172
Q0.4	Park_Light_console	0/1	055-173

Figure B.2: The second part of signals coming from the PBB PLC

PLC address	PLC tag / beschrijving	Type	Kanaal
Q0.6	Point_to_go_lamp	0/1	055-175
Q1.0	Cabin_Rot_L	0/1	055-177
Q1.1	Right_Canopy_Extend	0/1	055-178
Q1.2	Cabin_Rot_R	0/1	055-179
Q1.3	Right_Canopy_Retract	0/1	055-180
Q1.4	Left_Canopy_Extend	0/1	055-181
Q1.5	AL_Arm_Extend	0/1	055-182
Q1.6	Left_Canopy_Retract	0/1	055-183
Q1.7	AL_Arm_Retract	0/1	055-184
Q2.0	PBB_Fault	0/1	055-185
Q2.1	Roll_door_working	0/1	055-186
Q2.2	Roll_door_open	0/1	055-187
Q2.3	Buzzer	0/1	055-188
Q2.4	Roll_door_closed	0/1	055-189
Q2.5	Obstacle_light_cabin	0/1	055-190
Q2.6	Video_channel_change	0/1	055-191
Q2.7	Cabin_light	0/1	055-192
Q3.0	Video_camera_enable	0/1	055-193
Q3.1	Travel_light	0/1	055-194
Q3.2	Heater_glass_enable	0/1	055-195
Q3.4	Heater_Floor_enable	0/1	055-197
Q3.6	Parking_position	0/1	055-199
Q5.0	Alarm_Reset	0/1	055-209
Q5.1	Speed_change_gradually	0/1	055-210
Q5.2	Free_stop	0/1	055-211
Q5.3	Quick_stop	0/1	055-212
Q5.4	Left_Converter_run	0/1	055-213
Q5.5	Bypass_quick_stop	0/1	055-214
Q5.6	Right_Converter_run	0/1	055-215
Q6.0	Power_on	0/1	055-217
Q6.2	Bypass_Lamp	0/1	055-219
Q6.4	PLC_OK	0/1	055-221
Q6.6	Tunnel_Lighting	0/1	055-223
Q10.0	Stairdoor_Light	0/1	055-249
Q10.2	Gate_brand	0/1	055-251
Q12.0	VDGS_Lamp	0/1	056-010
Q12.1	Travel_light	0/1	056-011
Q12.2	Oil_pump_enable	0/1	056-012
Q12.4	Lift	0/1	056-014
Q12.6	Descent	0/1	056-016
IW31	Joystick_angle_Left_Right	Analoog Geheel getal (niet omgezet naar een werkelijke hoek)	
IW33	Joystick_angle_Forward_Backwards	Analoog Geheel getal (niet omgezet naar een werkelijke hoek)	
IW35	Cabin angle	Analoog Geheel getal (niet omgezet naar een werkelijke hoek)	
IW37	Height	Analoog Geheel getal (niet omgezet naar een werkelijke hoogte)	
IW39	Rotunda_Angle	Analoog Geheel getal (niet omgezet naar een werkelijke hoek)	
IW43	Bogie_Angle	Analoog Geheel getal (niet omgezet naar een werkelijke hoek)	
PID256	Length_1	Analoog Geheel getal (niet omgezet naar een werkelijke lengte)	
PID260	??	Onbekend / Niets	
PID264	Length_2	Analoog Geheel getal (niet omgezet naar een werkelijke lengte)	
QW214	Left_Wheel_Speed	Analoog Geheel getal (niet omgezet naar een werkelijke snelheid)	
QW216	Right_Wheel_Speed	Analoog Geheel getal (niet omgezet naar een werkelijke snelheid)	
PQD256	??	Onbekend / Niets	
PQD260	??	Onbekend / Niets	

Figure B.3: The third part of signals coming from the PBB PLC

C

Maintenance tickets analysis diagrams

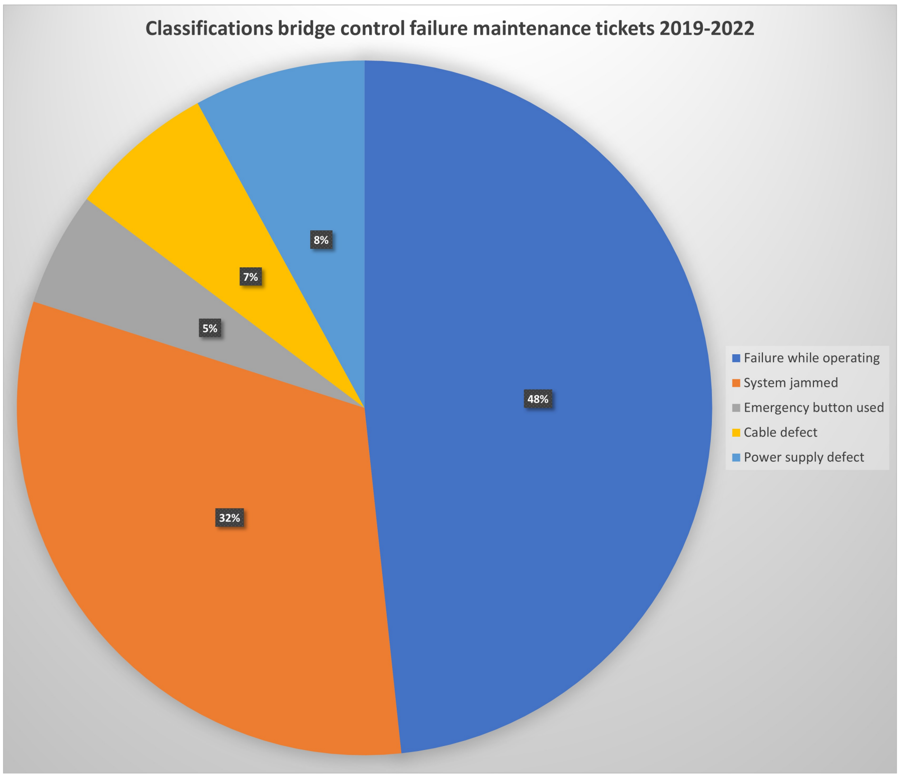


Figure C.1: Classifications bridge control failure maintenance tickets 2019-2022

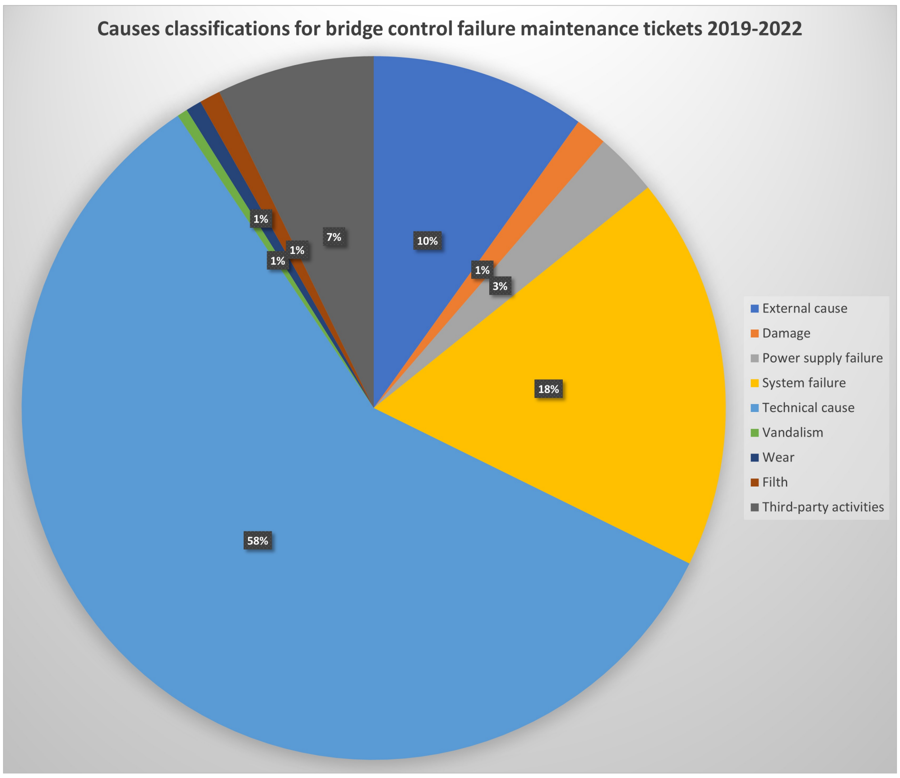


Figure C.2: Causes classifications for bridge control failure maintenance tickets 2019-2022

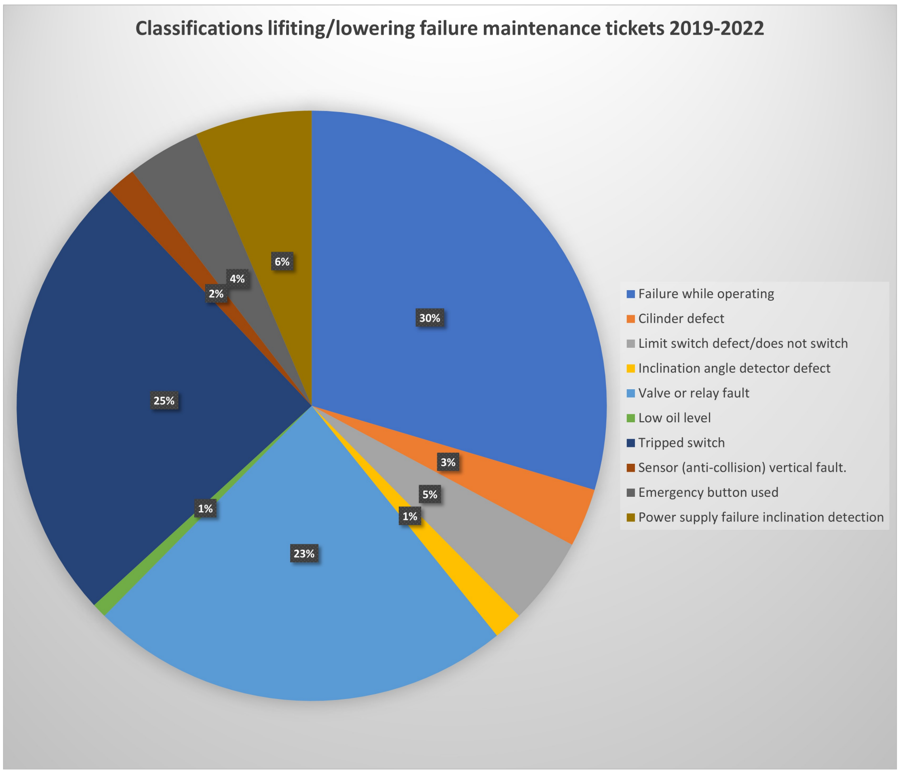


Figure C.3: Classifications lifting/lowering failure maintenance tickets 2019-2022

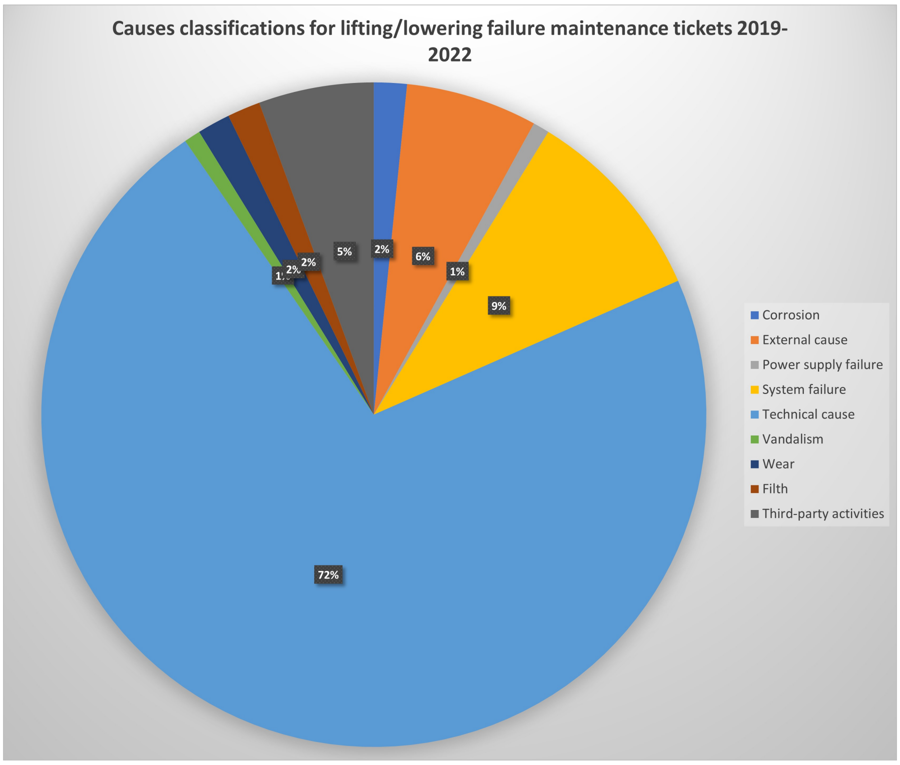


Figure C.4: Causes classifications for lifting/lowering failure maintenance tickets 2019-2022

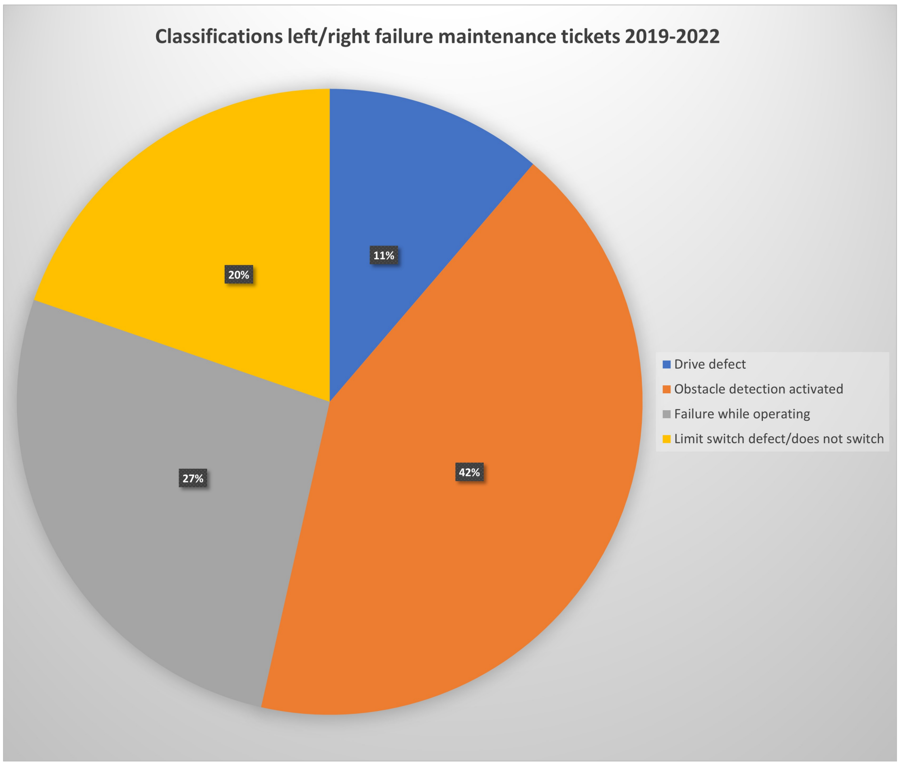


Figure C.5: Classifications left/right failure maintenance tickets 2019-2022

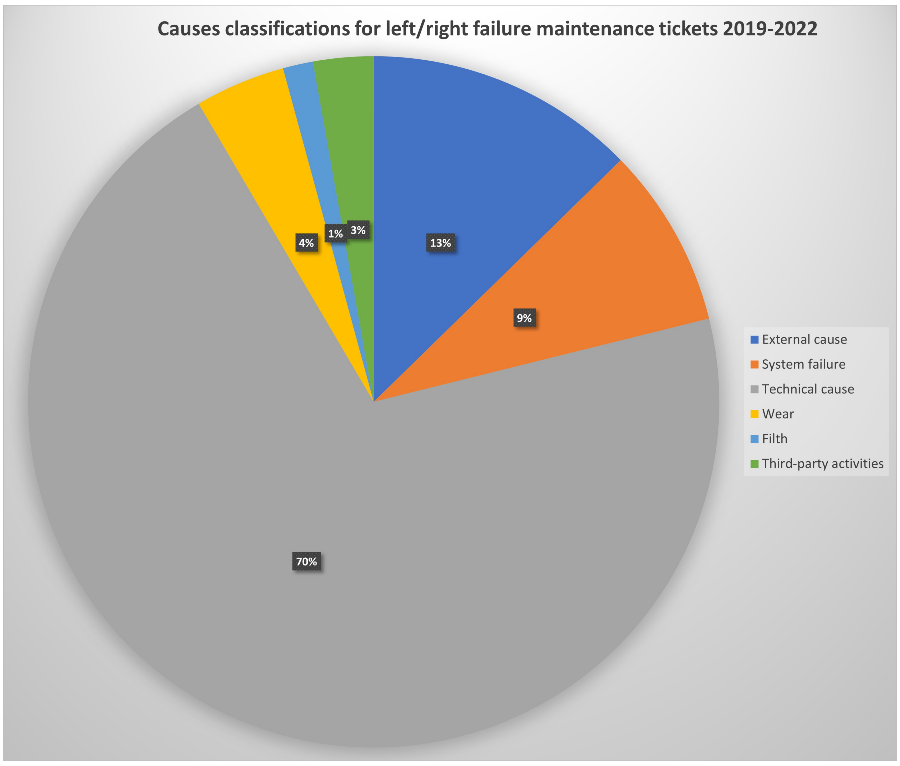


Figure C.6: Causes classifications for left/right failure maintenance tickets 2019-2022

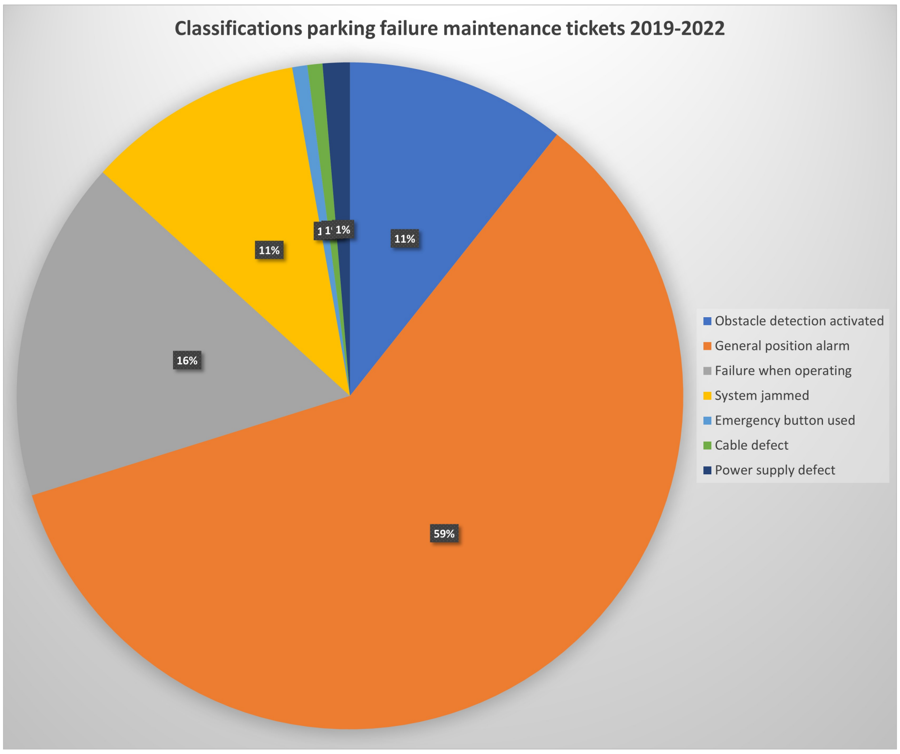


Figure C.7: Classifications parking failure maintenance tickets 2019-2022

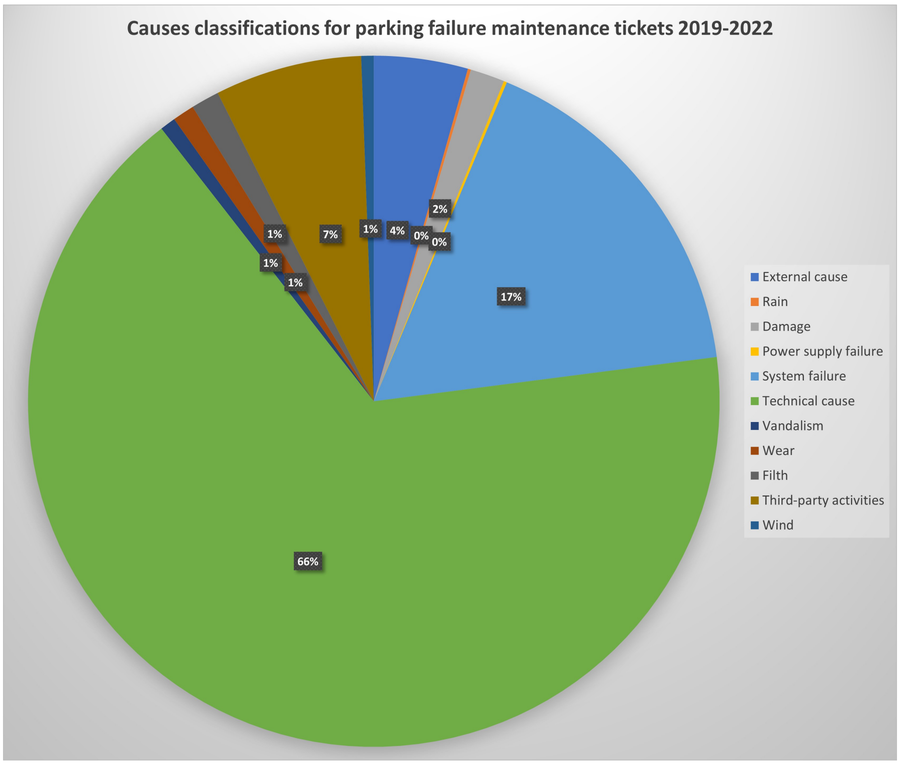


Figure C.8: Causes classifications for parking failure maintenance tickets 2019-2022

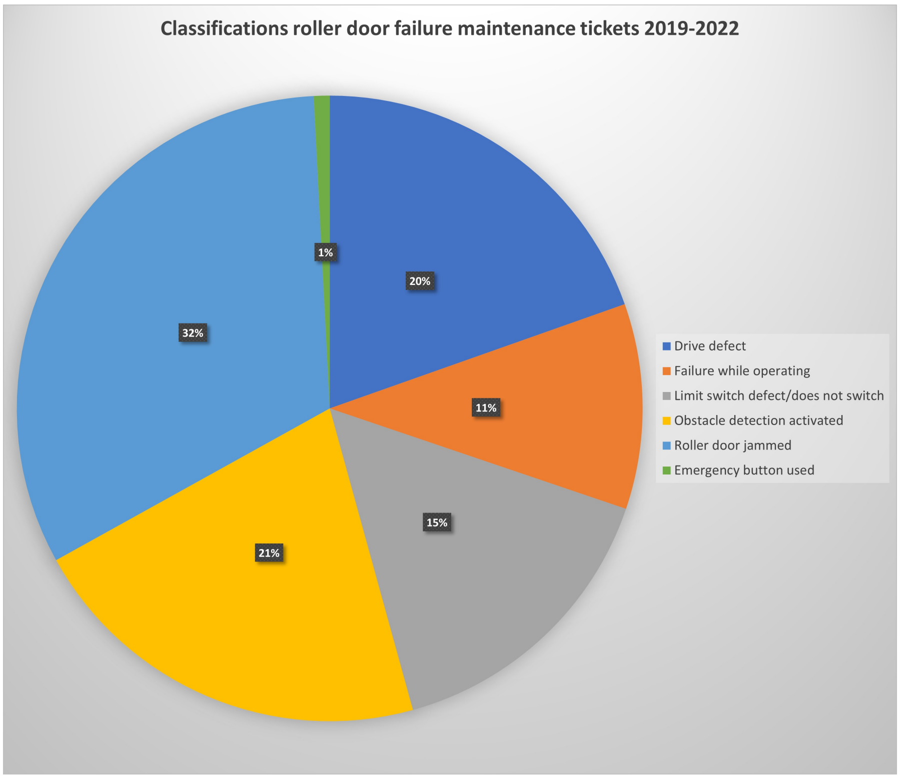


Figure C.9: Classifications roller door failure maintenance tickets 2019-2022

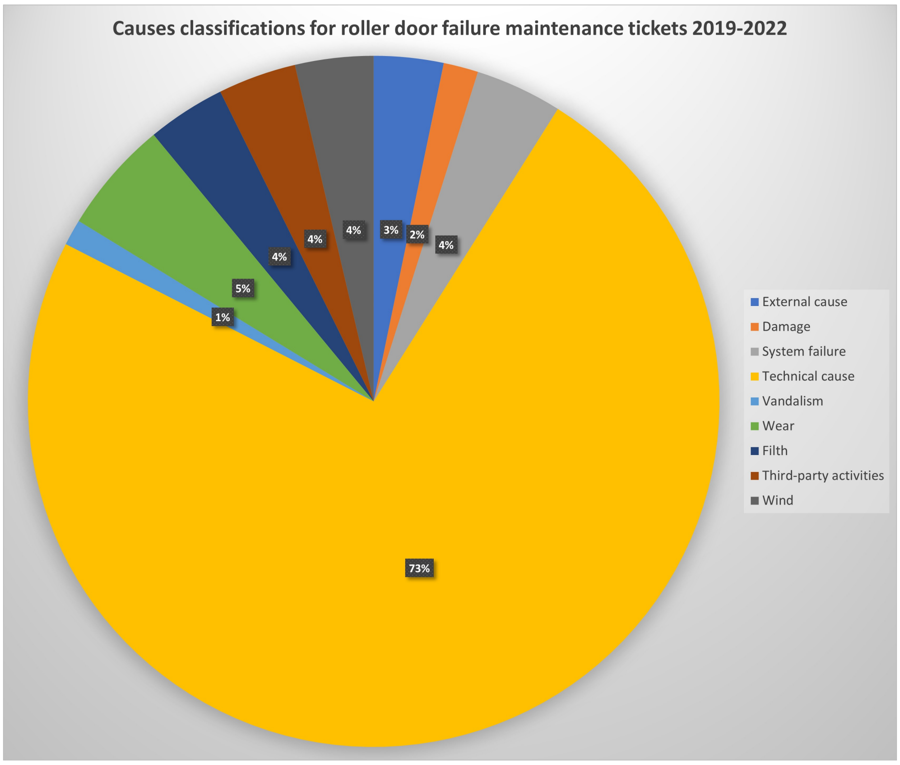


Figure C.10: Causes classifications for roller door failure maintenance tickets 2019-2022

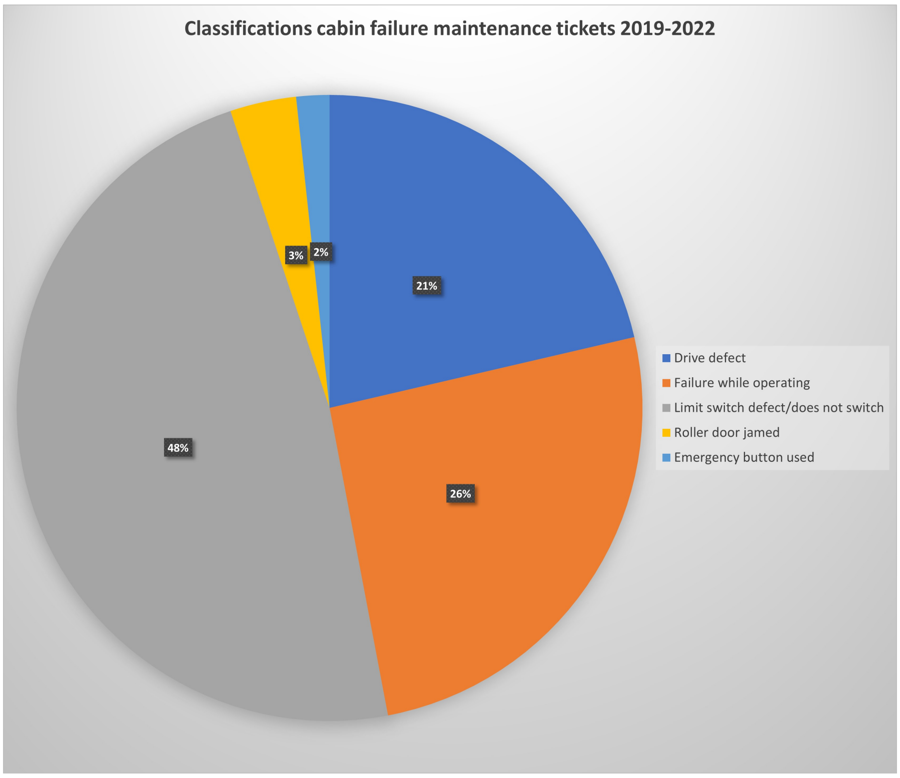


Figure C.11: Classifications cabin failure maintenance tickets 2019-2022

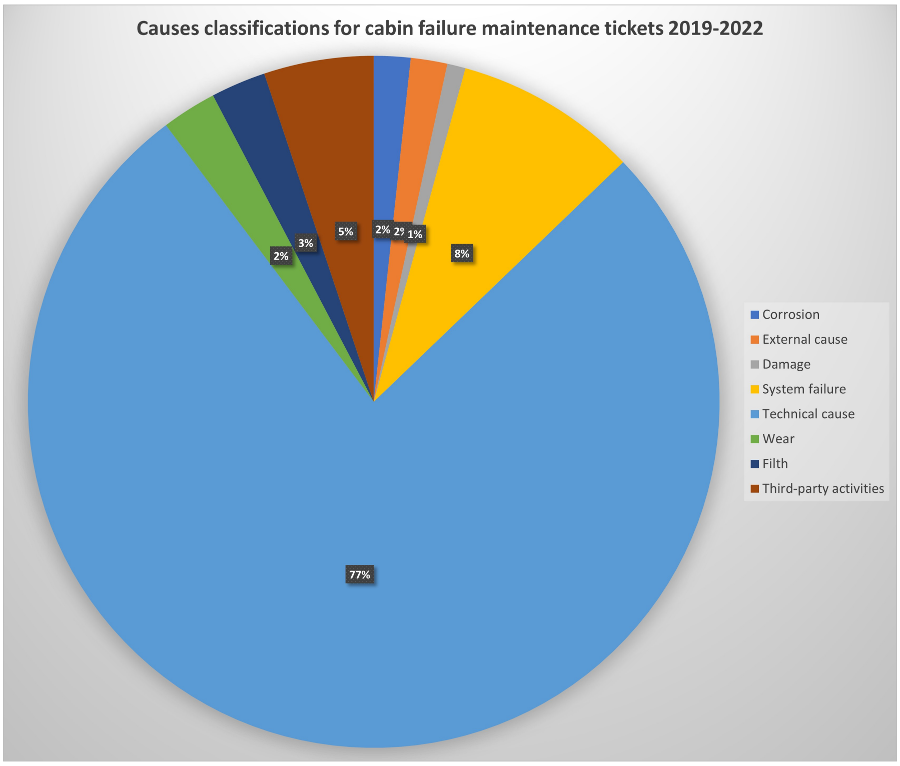


Figure C.12: Causes classifications for cabin failure maintenance tickets 2019-2022

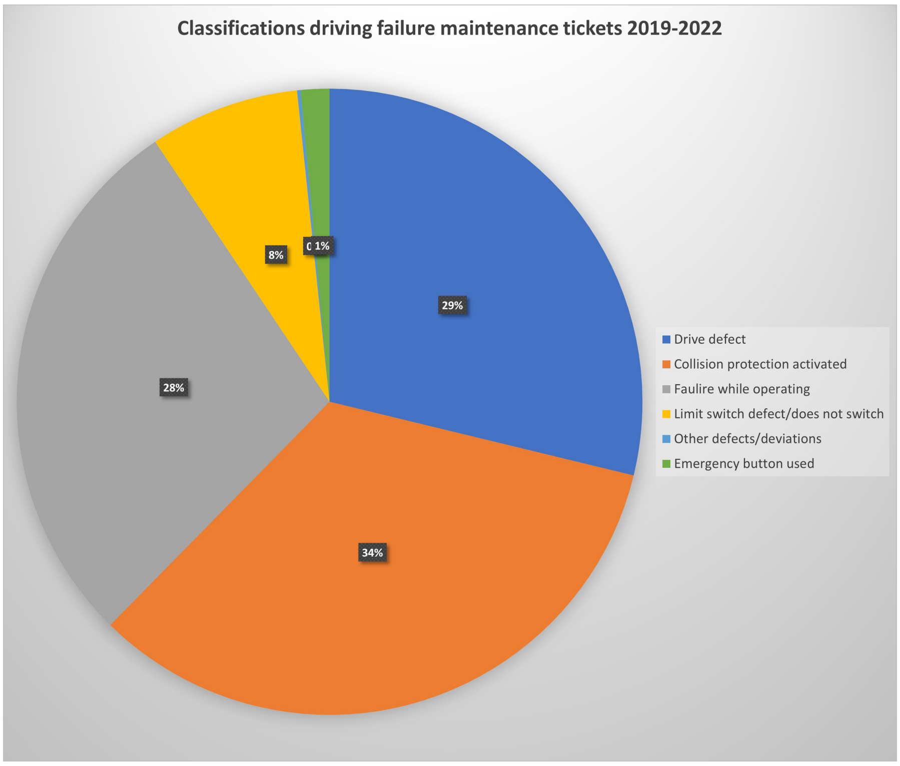


Figure C.13: Classifications driving failure maintenance tickets 2019-2022

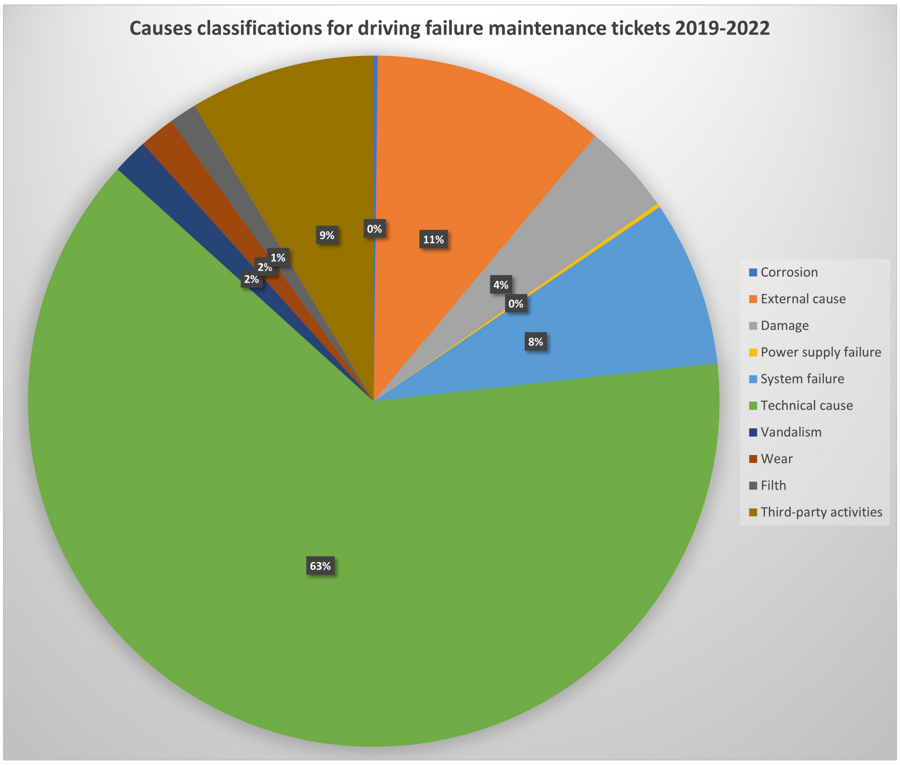


Figure C.14: Causes classifications for driving failure maintenance tickets 2019-2022

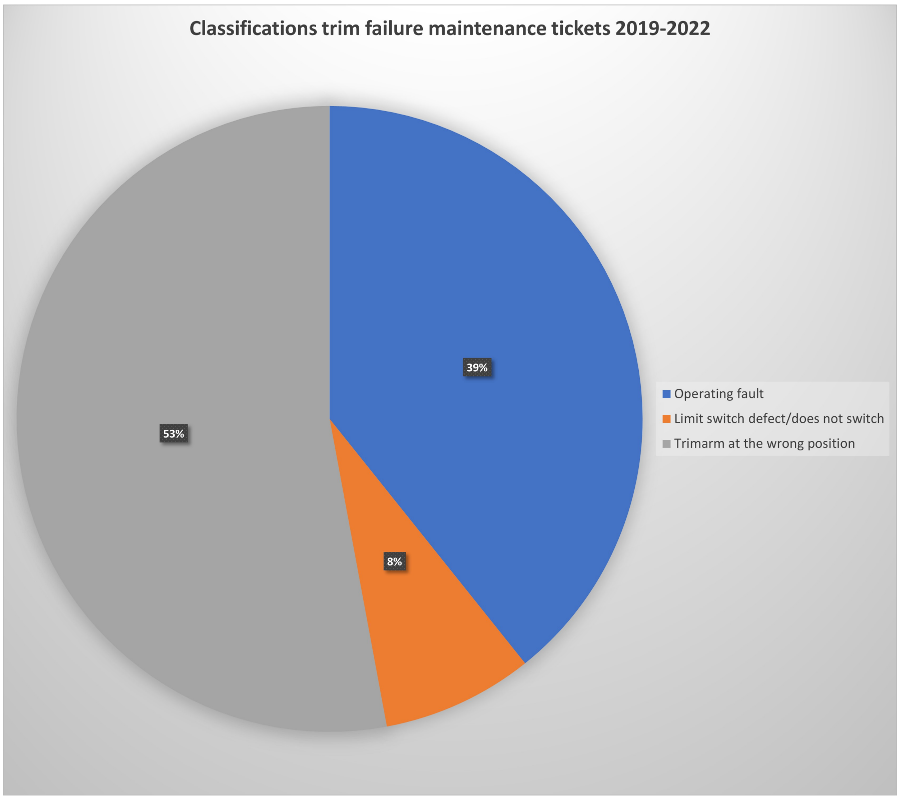


Figure C.15: Classifications trim failure maintenance tickets 2019-2022

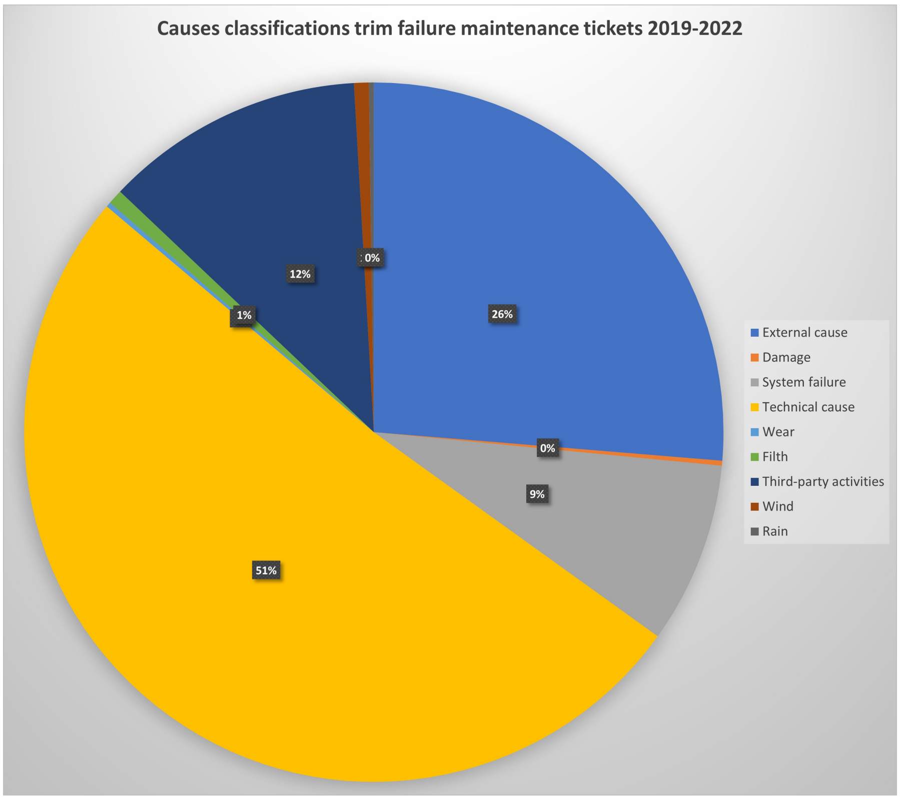


Figure C.16: Causes classifications for trim failure maintenance tickets 2019-2022

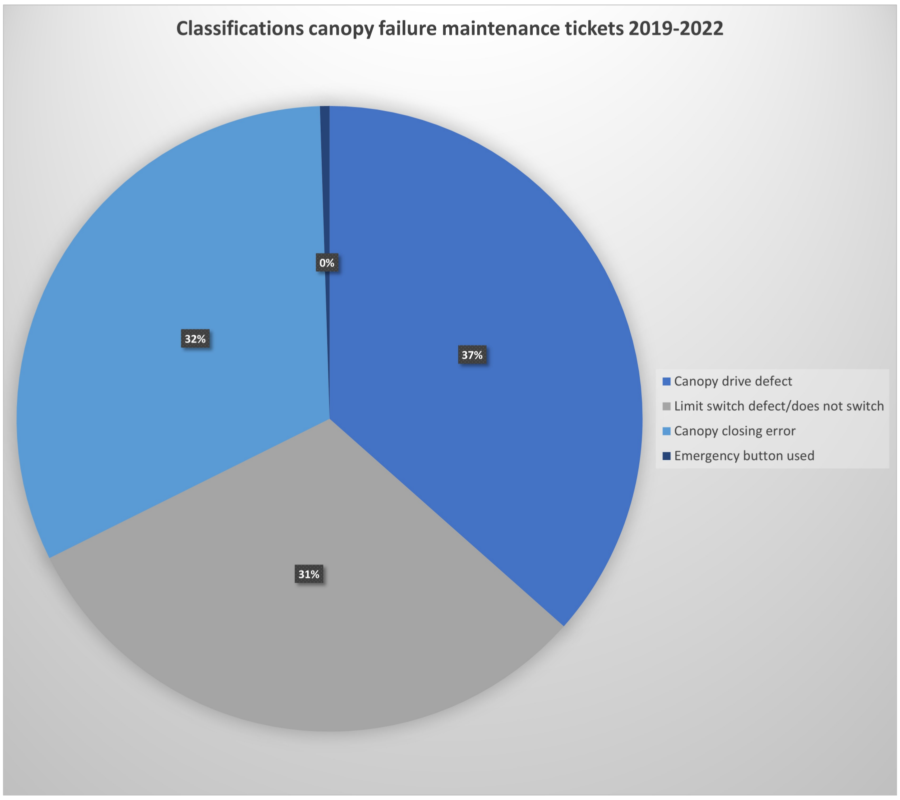


Figure C.17: Classifications canopy failure maintenance tickets 2019-2022

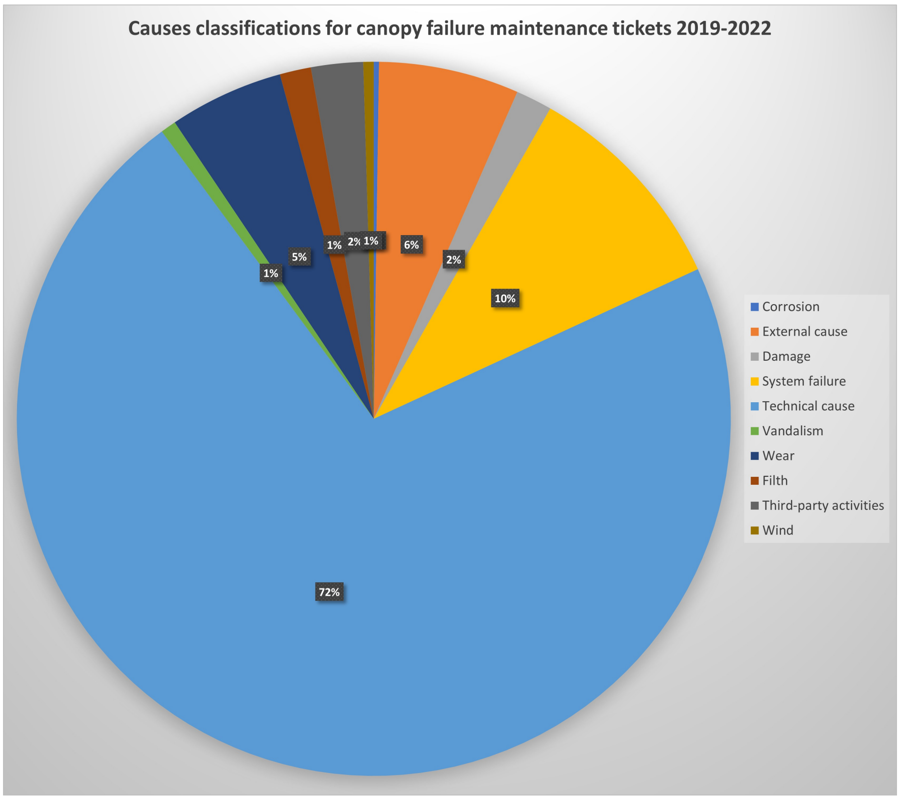


Figure C.18: Causes classifications for canopy failure maintenance tickets 2019-2022

D
Synthetic dataset

In this appendix, the synthetic dataset used in the simulation model is presented in table D.1 and D.2.

Table D.1: Sensor readings and corresponding failure probability (part 1)

Time step	Temperature in °C	Failure probability	Current in Ampere	Failure probability	Vibration in Hertz	Failure probability
1	27.08713233	2.54E-05	0.683094323	0.003879104	35.65476999	0.005910919
2	31.29942653	5.89E-05	0.83527992	0.015260603	34.23702076	0.007848725
3	29.35990782	3.99E-05	0.753296457	0.007296709	36.08368834	0.005424998
4	26.36139635	2.19E-05	0.872247799	0.021284635	34.17332849	0.007949345
5	24.99972531	1.67E-05	0.578880492	0.001518434	34.13932259	0.008003595
6	32.93876206	8.17E-05	0.614786936	0.002097692	34.18381391	0.007932692
7	31.35210458	5.95E-05	0.766284607	0.008201497	36.71669589	0.004779883
8	30.66458442	5.19E-05	0.643618137	0.002719148	35.04366258	0.006679364
9	29.85271963	4.41E-05	1.067834761	0.123750585	34.45648954	0.007511667
10	27.55197375	2.78E-05	0.672703665	0.003532791	35.01085644	0.006723333
11	31.49617252	6.12E-05	0.631314369	0.002434118	35.13101194	0.00656369
12	27.25440054	2.62E-05	0.797346956	0.010846884	34.83996448	0.006957098
13	26.28242499	2.16E-05	0.745047674	0.006774627	35.89330242	0.00563555
14	32.59202218	7.62E-05	0.613333528	0.002070431	34.12379596	0.008028487
15	30.78669952	5.31E-05	0.70620982	0.004776192	35.99604899	0.005520925
16	28.81487127	3.58E-05	0.693285194	0.004251713	33.23078776	0.009598413
17	30.90953791	5.45E-05	0.580908687	0.001546406	32.82743835	0.010404804
18	27.80701089	2.93E-05	0.765999663	0.008180492	35.37697852	0.006248613
19	28.94319921	3.68E-05	0.693118358	0.004245334	35.23659982	0.006426534
20	29.74384084	4.31E-05	0.698164475	0.004442581	35.42154793	0.006193161
21	31.9672783	6.73E-05	0.563347606	0.001320333	34.09059719	0.008081972
22	33.71918262	9.55E-05	0.819319064	0.013218625	34.46828026	0.007493974
23	30.50768597	5.03E-05	0.525062959	0.000935496	33.69242021	0.008751904
24	30.49033317	5.01E-05	0.726500733	0.005733131	35.53836429	0.006050146
25	30.80284737	5.33E-05	0.563340072	0.001320243	34.04969931	0.00814835
26	34.2333906	0.000105868	0.731955224	0.006021596	35.66578878	0.005897907
27	31.36617431	5.97E-05	0.82433034	0.013828454	32.58061094	0.010931331
28	30.83613414	5.37E-05	0.652525757	0.002946114	35.79125786	0.005751747
29	26.17260889	2.11E-05	0.814892847	0.012702399	35.08453272	0.006624989
30	31.52355761	6.16E-05	0.825567864	0.013983332	36.11528423	0.005390825
31	34.87047555	0.000120254	0.497096245	0.000727327	34.19755638	0.007910919
32	28.35849089	3.27E-05	0.591881164	0.001706912	34.8183961	0.006987173
33	33.47897344	9.10E-05	0.576125466	0.001481247	34.81791299	0.006987848
34	30.78004077	5.31E-05	0.805352885	0.011657284	34.16771714	0.007958272
35	29.22270727	3.89E-05	0.570186789	0.001404156	36.99444272	0.004521604
36	30.00900929	4.55E-05	0.721316287	0.005471768	34.81069225	0.006997947
37	30.65360543	5.17E-05	0.75984439	0.007739637	35.44047272	0.006169765
38	24.21322387	1.43E-05	0.746455652	0.00686102	33.61849985	0.008882254
39	30.5919686	5.11E-05	0.80516921	0.01163803	36.91834794	0.004590944
40	32.25984622	7.13E-05	0.642986657	0.002703738	35.30257019	0.006342298
41	32.99869771	8.27E-05	0.733448135	0.006103049	35.94298035	0.005579835
42	27.8319365	2.94E-05	0.598853455	0.001817454	35.47232395	0.006130587
43	31.92081757	6.67E-05	0.598672596	0.001814498	34.83974867	0.006957398
44	31.05075497	5.60E-05	0.392297102	0.000283209	36.464124	0.005027538
45	28.49264755	3.36E-05	0.684868559	0.003941543	34.10395852	0.008060403
46	31.84095121	6.56E-05	0.701811572	0.004590824	35.38585427	0.006237531
47	27.61483992	2.82E-05	0.826355752	0.01408284	36.70942898	0.004786835
48	29.328393	3.97E-05	0.77279858	0.00869669	35.39274333	0.006228943
49	29.75967233	4.33E-05	0.690959813	0.004163656	35.28051187	0.00637034

Table D.2: Sensor readings and corresponding failure probability (part 2)

Time step	Temperature in °C	Failure probability	Current in Ampere	Failure probability	Vibration in Hertz	Failure probability
50	30.24751753	4.77E-05	0.813644161	0.012560447	34.43929437	0.007537544
51	26.37519767	2.20E-05	0.779099534	0.009204119	34.65471018	0.007219699
52	32.34861012	7.26E-05	0.640004466	0.002632135	35.24745391	0.006412598
53	28.43226291	3.32E-05	0.695262996	0.004328072	35.15847981	0.00652773
54	31.80240095	6.51E-05	0.721401338	0.005475958	35.59765385	0.005978828
55	28.52100237	3.38E-05	0.800219991	0.011131013	35.69704713	0.00586115
56	29.52081981	4.13E-05	0.909529014	0.029770453	34.14037338	0.008001913
57	32.36144015	7.28E-05	0.739072019	0.006419906	34.25987709	0.007812928
58	29.02239164	3.73E-05	0.778731851	0.009173711	35.09033412	0.006617307
59	33.74091702	9.59E-05	0.688533346	0.004073715	34.61424663	0.007278363
60	32.75729813	7.88E-05	0.597936639	0.001802519	36.48247884	0.005009116
61	33.85720086	9.82E-05	0.565522249	0.001346428	33.85360705	0.008474265
62	28.82932138	3.59E-05	0.701329291	0.00457094	36.15344284	0.00534984
63	28.03200231	3.06E-05	0.660490819	0.003165062	35.37103534	0.006256045
64	28.30734733	3.24E-05	0.770193095	0.008495131	33.74577319	0.008659012
65	27.52972916	2.77E-05	0.828752001	0.014389853	34.45965429	0.007506914
66	32.35954883	7.28E-05	0.676835839	0.003666648	33.27454009	0.009514788
67	29.98175453	4.52E-05	0.590726973	0.001689273	35.31793462	0.006322839
68	29.63441928	4.22E-05	0.642864752	0.002700773	36.31769637	0.00517695
69	30.879586	5.41E-05	0.735759481	0.006231335	34.71472943	0.007133553
70	29.50488324	4.11E-05	0.539205656	0.00106248	34.70074289	0.007153535
71	28.64930833	3.47E-05	0.718542004	0.005336838	33.95655745	0.008301563
72	27.60995497	2.81E-05	0.764602374	0.008078261	36.32086189	0.005173673
73	28.68137501	3.49E-05	0.663715851	0.003258275	35.04606035	0.006676162
74	30.90610374	5.44E-05	0.554661776	0.00122105	36.16700226	0.005335352
75	31.29874609	5.89E-05	1.003899697	0.069606124	34.28216518	0.007778179
76	27.09356163	2.54E-05	0.532115994	0.000996804	35.85418162	0.005679817
77	28.64646133	3.46E-05	0.624039761	0.002279858	36.97388222	0.004540235
78	32.39590397	7.33E-05	0.688489677	0.004072114	34.31651596	0.007724925
79	32.72569852	7.83E-05	0.69879398	0.004467822	33.08802475	0.009876422
80	29.98923445	4.53E-05	0.760690839	0.007798823	33.66463496	0.008800674
81	32.93601631	8.17E-05	0.871010782	0.021048985	34.21093661	0.007889778
82	29.79486175	4.36E-05	0.820110587	0.013313127	32.91002206	0.010234362
83	29.44384079	4.06E-05	0.669015286	0.003417444	34.75128985	0.007081582
84	29.38001516	4.01E-05	0.597850189	0.001801117	35.1008032	0.006603466
85	32.67299547	7.75E-05	0.76569629	0.008158186	35.80469789	0.005736307
86	28.36141522	3.27E-05	0.801724664	0.011282775	33.7418026	0.008665891
87	26.93314409	2.46E-05	0.618359586	0.002166237	36.65445128	0.004839759
88	28.27105102	3.21E-05	0.850038017	0.017428337	35.26041236	0.006396
89	27.57738778	2.80E-05	0.753356357	0.007300644	34.61795217	0.007272971
90	32.00109515	6.77E-05	0.766569319	0.00822254	34.11661331	0.008040028
91	30.2418928	4.77E-05	0.702302021	0.004611132	34.71082013	0.007139132
92	28.53105351	3.38E-05	0.651634808	0.002922585	35.25117741	0.006407824
93	26.80064197	2.39E-05	0.783349391	0.009562984	34.32666412	0.007709262
94	30.62156466	5.14E-05	0.780880344	0.009352824	34.17522931	0.007946324
95	27.71665008	2.88E-05	0.55300355	0.001202962	34.78198384	0.007038243
96	32.88559796	8.09E-05	0.839362318	0.01583173	33.58898929	0.008934833
97	32.40445312	7.34E-05	0.634221433	0.002498644	35.12229366	0.006575144
98	33.42151417	9.00E-05	0.61590241	0.002118857	35.64573768	0.005921606
99	31.88673183	6.62E-05	0.610240572	0.002013592	34.51613644	0.00742259
100	30.65367361	5.17E-05	0.73474682	0.006174801	34.91998921	0.006846636

E

Maintenance logs analysis code

```
##libraries
import openai
import pandas as pd
import re
from time import time,sleep
from nltk.stem import WordNetLemmatizer

def open_file(filepath):
    with open(filepath, 'r', encoding='utf-8') as infile:
        return infile.read()

def save_file(filepath, content):
    with open(filepath, 'w', encoding='utf-8') as outfile:
        outfile.write(content)

# Load the excel file into a pandas DataFrame
df = pd.read_excel("preprocessed_data_new.xlsx")
df.head()

# Merge the columns "Oorzaak", "Oorzaak2", "Oorzaak3", and "Oorzaak4" into a single column "Oorzaak_merged"
df['Oorzaak_merged'] =  df[df.columns[6:10]].apply(
    lambda x: ' '.join(x.dropna().astype(str)),
    axis=1
)
#merged_text = df['Oorzaak_merged'].str.cat(sep=' ')
# Drop the original columns "Oorzaak", "Oorzaak2", "Oorzaak3", and "Oorzaak4"
df = df.drop(['Oorzaak', 'Oorzaak2', 'Oorzaak3', 'Oorzaak4'], axis=1)

# Make a new data set without the date of the maintenance ticket
data_set= df[["Geografische locatie omschrijving","Probleem (Storingsboom)", "Faalvorm (Storingsboom)", "Oorzaak (Storingsboom)",
"Herstelactie(Storingsboom)", "Oorzaak_merged"]]
data_set.head()


#Make a file containing stopwords source: https://github.com/lisanka93/text_analysis_python_101/blob/master/Railroad_incidents_USA2019.ipynb
stop_words_file = 'SmartStoplist.txt'

stop_words = []

with open(stop_words_file, "r") as f:
    for line in f:
        stop_words.extend(line.split())

stop_words = stop_words

# Filtering of the stopwords from the maintenance logs
def preprocess(raw_text):

    #regular expression keeping only letters - more on them later
    letters_only_text = re.sub("[^a-zA-Z]", " ", raw_text)

    # convert to lower case and split into words -> convert string into list ( 'hello world' -> ['hello', 'world'])
    words = letters_only_text.lower().split()

    cleaned_words = []
    lemmatizer = WordNetLemmatizer()

    # remove stopwords
    for word in words:
        if word not in stop_words:
            cleaned_words.append(word)

    #  #stemm or lemmatise words
    # stemmed_words = []
    # for word in cleaned_words:
    #     word = lemmatizer.lemmatize(word, pos=wordnet.NOUN)
    #     stemmed_words.append(word)

    # converting list back to string
    return " ".join(cleaned_words)

data_set['Oorzaak_merged'] = data_set['Oorzaak_merged'].apply(preprocess)

#Make a new data set to test openAI COT prompts on the Huif storing classification
storing_set = data_set.loc[(data_set['Probleem (Storingsboom)'] == 'Huif storing') & (data_set['Faalvorm (Storingsboom)'] == 'Eindschakelaar defect/schakelt
niet')].copy()
maintenance_logs = storing_set[['Oorzaak_merged']].copy()
maintenance_logs = maintenance_logs.reset_index()
maintenance_logs = maintenance_logs[['Oorzaak_merged']].copy()
subset_logs = maintenance_logs.iloc[75:130]
```

```
textblock ='\n'.join(subset_logs['Oorzaak_merged'])

openai.api_type = "azure"
openai.api_base = "https://open-ai-pilot.openai.azure.com/"
openai.api_version = "2022-12-01"
openai.api_key = open_file('openaiapikey.txt')

def ai_implementation(prompt, engine= 'chat-gpt-base', temp=0.1, top_p=1, tokens=3000):
    max_retry = 1
    retry = 0
    prompt = prompt.encode(encoding='ASCII',errors='ignore').decode()
    while True:
        try:
            response = openai.Completion.create(
                engine=engine,
                prompt=prompt,
                temperature=temp,
                max_tokens=tokens,
                top_p=top_p,
                )
            text = response['choices'][0]['text'].strip()
            #text = re.sub('\s+', ' ', text)
            filename = 'switch_2.txt'
            save_file('Huif/%s' % filename, prompt + '\n\n==========\n\n' + text)
            return text
        except Exception as oops:
            retry += 1
            if retry >= max_retry:
                return "AI error: %s" % oops
            print('Error communicating with OpenAI:', oops)
            sleep(1)

# Replace the <<logs>> placeholder in the prompt file with textblock
prompt_file = 'prompt_one.txt'
with open(prompt_file, "r") as f:
    prompt = f.read().replace('<<logs>>', textblock)

response = ai_implementation(prompt)
```

F

Decision-making code

```
# Libraries
import gurobipy as gp
from gurobipy import GRB
import matplotlib.pyplot as plt
import random
import pandas as pd
import numpy as np

# ---- General Parameters ----
np.random.seed(2)
components = [1, 2, 3]                                           # Definition of the components
df = pd.read_csv('base.csv',delimiter=';', header=None)          # Load failure probabilities in a pandas dataframe
CRT_shop = [0, 0, 0]                                             # Corrective repair time for component i at the workshop
PRT_shop = [0, 0, 0]                                             # Proactive repair time for component i at the workshop
AT_value = 15                                                    # The arrival time of a mechanic
time_step = list(range(1, 451))                                  # Time step t
mechanics = [random.choice([1,1]) for _ in range(450)]           # The amount of mechanics at time step t
availability = [np.random.normal(loc=76, scale=54)for _ in range(450)]   # The availability of the aircraft stand to do maintenance activities
availability = [max(0, value) for value in availability]        # If the availability is lower than zero, set zero
alpha = 1                                                        # Reduction coefficient alpha
beta = 1                                                         # Reduction coefficient beta
reliability = [0.1, 0.1, 0.1]                                    # The reliability treshold for component i

# ---- Failure probabilities----
F_canopy = df.iloc[0, :].tolist()
F_gas = df.iloc[1, :].tolist()                                   # Set the first row of df as failure probability of the gas spring
F_drive= df.iloc[2, :].tolist()                                  # Set the second row of df as failure probability of the drive
F_limit= df.iloc[3, :].tolist()                                  # Set the third row of df as failure probability of the limit switch
F = [F_drive, F_limit, F_gas]
F_adap = [F_drive, F_limit, F_gas,F_canopy]
num_timesteps = len(time_step)
num_components = len(components)

# Make CRT and PRT matrices with zeros
CRT_site = np.zeros((num_timesteps, num_components))
PRT_site = np.zeros((num_timesteps, num_components))

for t in range(num_timesteps):
    # Generate random values for d, l, and g for each time step
    d = np.random.gamma(shape=0.6810321708199347, scale=95.1056707187164)
    l = np.random.lognormal(mean=np.log(38.01132442922867), sigma=0.7863368878972384)
    g = np.random.gamma(shape=3.7916764151620947, scale=9.075371045719955)

    # Replace the zeros with the actual values
    CRT_site[t] = [d, l, g]
    PRT_site[t] = [d/2, l / 2, g / 2]
```

Local strategy

```
def calculate_AR(AT_value, time_step):                           # Definition of the arrival rate
    AR = []                                                      # List to store the calculated arrival rate
    for t in range(len(time_step)):                              # Iterate over each time step
        AR_v = AT_value / time_step[t]                           # Calculate the arrival rate for each timestep
        AR.append(AR_v)                                          # Add the arrival time rate for the specific time step to the list
    return AR                                                    # Return the list of AR values

def calculate_CRT(time_step, components, CRT_site, CRT_shop, alpha, beta, mechanics):        # Definition of the corrective repair time
    CRT = []                                                     # List to store the calculated corrective repair
    for t in range(len(time_step)):                              # Iterate over each time step
        CRT_time_step = []                                       # List to store the calculated corrective repair time for each time step
        for i in range(len(components)):                         # Iterate over each component
            CRT_value = (CRT_site[t][i] / (alpha * mechanics[t])) + (CRT_shop[i] / (beta * mechanics[t]))     # Calculate the corrective repair time for component i at
timestep t
            CRT_time_step.append(CRT_value)                      # Add the corrective repair time of component i at timestep t to the list
        CRT.append(CRT_time_step)                                # Add the corrective repair time to the list
    return CRT                                                   # Return the list of CRT values

def calculate_PRT(time_step, components, PRT_site, PRT_shop, alpha, beta, mechanics):        # Definition of the proactive repair time
    PRT = []                                                     # List to store the calculated proactive repair time
    for t in range(len(time_step)):                              # Iterate over each time step
        PRT_time_step = []                                       # List to store the calculated proactive repair time for each time step
        for i in range(len(components)):                         # Iterate over each component
            PRT_value = (PRT_site[t][i] / (alpha * mechanics[t])) + (PRT_shop[i] / (beta * mechanics[t]))     # Calculate the proactive repair time for component i at
timestep t
            PRT_time_step.append(PRT_value)                      # Add the proactive repair time of component i at timestep t to the list
        PRT.append(PRT_time_step)                                # Add the proactive repair time to the list
    return PRT                                                   # Return the list of PRT values

def calculate_RR(time_step, F, CRT, PRT):                        # Definition of the repair rate
    RR = []                                                      # List to store the calculated repair rate
```

```
    for t in range(len(time_step)):                                        # Iterate over each time step
        RR_time_step = []                                                  # List to store the calculated repair rate for each time step
        for i in range(len(components)):                                   # Iterate over each component
            RR_value = ((CRT[t][i] * (F[i][t])) + (PRT[t][i] * (1-F[i][t]))) / time_step[t]     # Calculate the repair rate for component i at timestep t
            RR_time_step.append(RR_value)                                  # Add the repair rate of component i at timestep t to the list
        RR.append(RR_time_step)                                            # Add the repair rates to the list
    return RR                                                              # Return the list of RR values

def calculate_TR(RR, AR,components):                                       # Definition of the total repair rate
    TR = []                                                                # List to store the calculated total repair rate
    for t in range(len(time_step)):                                        # Iterate over each time step
        TR_time_step = []                                                  # List to store the calculated total repair rate for each time step
        for i in range(len(components)):                                   # Iterate over each component
            TR_value = RR[t][i] + AR[t]                                    # Calculate the total repair rate for component i at timestep t
            TR_time_step.append(TR_value)                                  # Add the total repair rate of component i at timestep t to the list
        TR.append(TR_time_step)                                            # Add the total repair rates to the list
    return TR                                                              # Return the list of TR values

def calculate_repairtime(time_step,PRT, AT_value):                         # Definition of the actual total repair time
    repairtime=[]                                                          # List to store the calculated repair time
    for t in range(len(time_step)):                                        # Iterate over each time step
        repairtime_time_step =[]                                           # List to store the calculated repair time for each time step
        for i in range(len(components)):                                   # Iterate over each component
            repeartime_value = PRT[t][i] + AT_value                        # Add to the proactive repair time the arrival time
            repairtime_time_step.append(repeartime_value)                  # Add the repair time of component i at timestep t to the list
        repairtime.append(repairtime_time_step)                            # Add the repair time to the list
    return repairtime                                                      # Return the list of repair time values

# Call the functions to get the results
AR= calculate_AR(AT_value, time_step)
CRT = calculate_CRT(time_step, components, CRT_site, CRT_shop, alpha, beta, mechanics)
PRT = calculate_PRT(time_step, components, PRT_site, PRT_shop, alpha, beta, mechanics)
RR= calculate_RR(time_step, F, CRT, PRT)
TR= calculate_TR(RR, AR,components)
repairtime= calculate_repairtime(time_step,PRT,AT_value)

# Convert component indices to strings for using in the loop
component_names = list(map(str, components))

# Create a dictionary to store the optimal time steps for each component individually
optimal_time_steps = {}
t_opt = {}

# Loop over all components to get the optimal time step
for i in range(len(components)):
    row = F[i]
    if any(element > reliability[i] for element in row):
        model = gp.Model("time_step_minimization_" + component_names[i])

        # Add variables time_step[t] for component [i]
        time_step_vars = model.addVars(len(time_step), lb=0, vtype=GRB.BINARY, name="time_step_" + component_names[i])

        # Add binary variables to indicate if maintenance is scheduled at time step t
        maintenance_scheduled = model.addVars(len(time_step), vtype=GRB.BINARY, name="maintenance_scheduled_" + component_names[i])

        # Add objective function to minimize total repair time for component [i]
        model.setObjective(gp.quicksum(TR[t][i] * time_step_vars[t] for t in range(len(time_step))), GRB.MINIMIZE)

        # Add constraint sum of time_step[t] for this component == 1
        model.addConstr(time_step_vars.sum() == 1, name="time_step_sum_constraint_" + component_names[i])

        # Add constraint for reliability threshold of component[i]
        for t in range(len(time_step)):
            model.addConstr(reliability[i] >= time_step_vars[t] * F[i][t], name="reliability_constraint_" + component_names[i])

        # Add constraint to check if maintenance time does not exceed availability
        for t in range(len(time_step)):
            # maintenance_scheduled[t] is 1 if maintenance is scheduled at time step t
            # Ensure that maintenance_scheduled[t] is greater than or equal to time_step_vars[t]
            model.addConstr(maintenance_scheduled[t] >= time_step_vars[t], name="maintenance_scheduled_constraint_" + component_names[i])

            # The following constraint checks if maintenance time does not exceed availability
            # Use a big M (M_big) to represent an upper bound on maintenance time
            M_big = max(availability) * 2
            model.addConstr(repairtime[t][i] - availability[t] * maintenance_scheduled[t] <= M_big * (1 - time_step_vars[t]), name="maintenance_time_constraint_"
+ component_names[i])

        # Optimize the model
        model.optimize()

        # Check if the optimization is successful
```

```
        if model.status == GRB.OPTIMAL:
            # Retrieve the optimal variable values for this component
            optimal_vars = [time_step_vars[t].x for t in range(len(time_step))]
            # Find the index of the optimal variable (time step)
            optimal_time_step_index = optimal_vars.index(1)
            # Store the optimal time step for this component
            t_opt[component_names[i]] = time_step[optimal_time_step_index]
            t_opt= {int(key): value for key, value in t_opt.items()}
        else:
            print("Optimization failed for component", component_names[i]+ ". Availibility schedule does not allow maintenance", ". Gurobi status:", model.status)
    else:
        print("Component " + component_names[i]+ " will not be above reliability treshhold this time period")

# Print the optimal time steps
for component, optimal_step in t_opt.items():
    print("Optimal time step for", component, ":", optimal_step)
```

Global strategy

```
def calculate_AR_group(AT_value, time_step):                # Definition of the group arrival rate
    AR_group = []                                           # List to store the calculated group arrival rate
    for t in range(len(time_step)):                         # Iterate over each time step
        AR_v_group = AT_value / time_step[t]                # Calculate arrival rate for a group
        AR_group.append(AR_v_group)                         # Add arrival rate to the group list
    return AR_group                                         # Return the list of AR values for each group

def calculate_RR_group(groups, time_step, F, CRT, PRT):     # Definition of the group repair rate
    RR_group = [[] for _ in range(len(groups))]             # List to store calculated repair rate for each group
    for t in range(len(time_step)):                         # Iterate over each time step
        for g, group_indices in enumerate(groups):          # Iterate over each group
            RR_sum = sum(((CRT[t][i-1] * F[i-1][t]) + PRT[t][i-1] * (1 - F[i-1][t])) for i in group_indices) / time_step[t]        # Calculate the sum of CRT and PRT values for components in the group
            RR_group[g].append(RR_sum)                      # Add to the group's list
    return RR_group                                         # Return the list of RR values for each group

def calculate_TR_group(RR_group, AR_group, time_step):      # Definition of the group total rate
    TR_group = []                                           # List to store the calculated group total rate
    for t in range(len(time_step)):                         # Iterate over each time step
        TR_value_group = RR_group[t] + AR_group[t]          # Calculate total repair rate for a group
        TR_group.append(TR_value_group)                     # Add total repair rate to the group list
    return TR_group                                         # Return the list of TR values for each group


def calculate_Repairtime_group(time_step, PRT, AT_value, groups):    # Definition of the total system repair time
    repairtime_group = [[] for _ in range(len(groups))]              # List to store calculated repair time for each group
    for t in range(len(time_step)):                                  # Iterate over each time step
        for g, group_index in enumerate(groups):                     # Iterate over each group
            repairtime_group_sum = sum(PRT[t][i-1] for i in group_index) + AT_value    # Calculate total repair time for a group
            repairtime_group[g].append(repairtime_group_sum)         # Add to the group's list
    return repairtime_group                                          # Return the list of repair times for each group

def calculate_optimal_group(t_opt, F, CRT, PRT, TR, AT_value):
    sorted_t_opt = sorted(t_opt.items(), key=lambda x: x[1])
    sorted_components = [item[0] for item in sorted_t_opt]
    if len(sorted_components) >1:
        def calculate_optimal_maintenance_time(time_step, t_opt, F, CRT, PRT, TR, AT_value):
            # Create a sorted list of component indices based on t_opt values
            sorted_t_opt = sorted(t_opt.items(), key=lambda x: x[1])
            sorted_components = [item[0] for item in sorted_t_opt]
            value_2 = sorted_t_opt[1][1]
            value_3 = sorted_t_opt[2][1]
            # Initialize variables
            index = 0

            # Start forming a new group
            component = sorted_components[index]
            group = [component]
            group_RR = calculate_RR_group([group], time_step, F, CRT, PRT)[0]       # Calculate RR for the group
            group_AR = calculate_AR_group(AT_value, time_step)                      # Use the same AT_value for the entire group
            components_outside = [comp for comp in sorted_components if comp not in group]
            # Calculate TT_group for the initial component
            TR_group = calculate_TR_group(group_RR, group_AR, time_step)
            #The actual repair time
            repairtime_group = calculate_Repairtime_group(time_step,PRT,AT_value,[group])[0]

            # Make the gurobi model
            model = gp.Model("time_step_minimization_group")
```

```
        # Add variables time_step[t] for component [i]
        time_step_group_vars = model.addVars(len(time_step), lb=0, vtype=GRB.BINARY, name="time_step_group")

        # Add binary variables to indicate if maintenance is scheduled at time step t
        maintenance_scheduled_group = model.addVars(len(time_step), vtype=GRB.BINARY, name="maintenance_scheduled_group")

        # Add objective function to minimize total repair time for component [i]
        model.setObjective(gp.quicksum(TR_group[t] * time_step_group_vars[t] for t in range(len(time_step))), GRB.MINIMIZE)

        # Add constraint sum of time_step[t] for this component == 1
        model.addConstr(time_step_group_vars.sum() == 1, name="time_step_sum_constraint_group")

        # Add constraint to check if maintenance time does not exceed availability
        for t in range(len(time_step)):
          # maintenance_scheduled[t] is 1 if maintenance is scheduled at time step t
          # Ensure that maintenance_scheduled[t] is greater than or equal to time_step_vars[t]
          model.addConstr(maintenance_scheduled_group[t] >= time_step_group_vars[t], name="maintenance_scheduled_constraint_group")

          # The following constraint checks if maintenance time does not exceed availability
          # Use a big M (M_big) to represent an upper bound on maintenance time
          M_big = max(availability) * 2
          model.addConstr(repairtime_group[t] - availability[t] * maintenance_scheduled_group[t] <= M_big * (1 - time_step_group_vars[t]),
    name="maintenance_time_constraint_group")


        # Add contraint for reliability threshold of component[i]
        for t in range(len(time_step)):
          model.addConstr(time_step_group_vars[t] * F[component-1][t] <= reliability[component-1], name="reliability_constraint_group")

        # Optimize the model
        model.optimize()

        # Check if the optimization is successful
        if model.status == GRB.OPTIMAL:
          # Retrieve the optimal variable values for this component
          optimal_group_vars = [time_step_group_vars[t].x for t in range(len(time_step))]
          # Find the index of the optimal variable (time step)
          optimal_time_step_index = optimal_group_vars.index(1)
          # Store the optimal time step for this component
          t_opt_group = time_step[optimal_time_step_index]
          TR_sys = TR_group[t_opt_group-1] + TR[value_2-1][sorted_components[1]-1] + TR[value_3-1][sorted_components[2]-1]
          print("TR_sys", TR_sys)
          index_2= 1
          while index_2 < len(sorted_components):
            #Choose the next component for the group
            j = sorted_components[index + 1]
            # Calculate TT_group for the new component
            new_group = group + [j]
            print("new_group =",new_group)
            print("old group was = ", group)
            new_group_RR = calculate_RR_group([new_group], time_step, F, CRT, PRT)[0]
            new_TR_group = calculate_TR_group(new_group_RR, group_AR, time_step)
            components_outside_new =[comp for comp in sorted_components if comp not in new_group]

            # Actual repair time for the existing group and the new group
            repairtime_group = calculate_Repairtime_group(time_step,PRT,AT_value,[group])[0]
            repairtime_group_new = calculate_Repairtime_group(time_step,PRT,AT_value,[new_group])[0]

            # Make the gurobi model
            model = gp.Model("time_step_minimization_group_new")

            # Add variables time_step[t] for component [i]
            time_step_group_vars_new = model.addVars(len(time_step), lb=0, vtype=GRB.BINARY, name="time_step_new")

            # Add binary variables to indicate if maintenance is scheduled at time step t
            maintenance_scheduled_group_new = model.addVars(len(time_step), vtype=GRB.BINARY, name="maintenance_scheduled_group_new")

            # Add objective function to minimize total repair time for component [i]
            model.setObjective(gp.quicksum(new_TR_group[t] * time_step_group_vars_new[t] for t in range(len(time_step))), GRB.MINIMIZE)

            # Add constraint sum of time_step[t] for this component == 1
            model.addConstr(time_step_group_vars_new.sum() == 1, name="time_step_sum_constraint_group_new")

            # Add constraint to check if maintenance time does not exceed availability
            for t in range(len(time_step)):
              # maintenance_scheduled[t] is 1 if maintenance is scheduled at time step t
              # Ensure that maintenance_scheduled[t] is greater than or equal to time_step_vars[t]
              model.addConstr(maintenance_scheduled_group_new[t] >= time_step_group_vars_new[t],
    name="maintenance_scheduled_constraint_group_new")

              # The following constraint checks if maintenance time does not exceed availability
```

```
                # Use a big M (M_big) to represent an upper bound on maintenance time
                M_big = max(availability) * 2
                model.addConstr(repairtime_group_new[t] - availability[t] * maintenance_scheduled_group_new[t] <= M_big * (1 -
time_step_group_vars_new[t]), name="maintenance_time_constraint_group_new")
            # # Add contraint for reliability threshold of component[i]
            for i in new_group:
                for t in range(len(time_step)):
                    model.addConstr(time_step_group_vars_new[t] * F[i-1][t] <= reliability[i-1], name=f"reliability_constraint_group_new_{i}")
            # Optimize the model
            model.optimize()

            # Check if the optimization is successful
            if model.status == GRB.OPTIMAL:
                # Retrieve the optimal variable values for this component
                optimal_group_vars_new = [time_step_group_vars_new[t].x for t in range(len(time_step))]
                # Find the index of the optimal variable (time step)
                optimal_time_step_index_new = optimal_group_vars_new.index(1)
                # Store the optimal time step for this component
                t_opt_group_new = time_step[optimal_time_step_index_new]
                if len(new_group) < 3:
                    TR_sys_new = new_TR_group[t_opt_group_new-1]+ TR[value_3-1][sorted_components[2]-1]
                    print("TR_sys for group of 2 components is: ", TR_sys_new)
                else:
                    TR_sys_final = TR_group[t_opt_group_new-1]
                    print("TR_sys for group of 3 components is: ", TR_sys_final)
                # If the new TT_sys is less than the existing one, update the group
                if TR_sys > TR_sys_new:
                    group = new_group
                    group_RR = new_group_RR
                    TR_group = new_TR_group
                    components_outside = components_outside_new
                    repairtime_group = repairtime_group_new
                    index += 1
                    index_2 +=1
                else:
                    new_group = group
                    components_outside_new = components_outside
                    repairtime_group_new = repairtime_group
                    index_2 = len(sorted_components)
                    print("The new group has a total maintenance rate for the system which is higher than the total maintenance rate of the system based on the
old group. Therefore no more calculations will be done.")
            else:
                print("error")
        elif model.status == GRB.INFEASIBLE:
            print("\n")
            print("No groups can be formed due to unavailability of the site.")
            print("Maintain each component individually.")
            new_group=[]
            components_outside_new = []
            t_opt_group_new = []
        return new_group, components_outside_new,sorted_components, t_opt_group_new

        optimal_group, individually_maintained,sorted_components, t_opt_group_new = calculate_optimal_maintenance_time(time_step, t_opt=t_opt, F=F,
CRT=CRT, PRT=PRT, TR=TR, AT_value=AT_value)
    else:
        print("Only one component is maintained this time period, so no groups can be formed")
    return optimal_group, individually_maintained,sorted_components, t_opt_group_new

optimal_group, individually_maintained,sorted_components, t_opt_group_new = calculate_optimal_group(t_opt, F, CRT, PRT, TR, AT_value)
```

Global strategy adjusted for group limit switch and gas spring

```
def calculate_optimal_group_individual(time_step, t_opt, F, CRT, PRT, TR, AT_value):
    sorted_t_opt = sorted(t_opt.items(), key=lambda x: x[1])
    sorted_components = [item[0] for item in sorted_t_opt]
    value_2 = sorted_t_opt[1][1]
    value_3 = sorted_t_opt[2][1]
    # Initialize variables
    index = 0

    # Start forming a new group
    component = sorted_components[index]
    group = [component]
    group_RR = calculate_RR_group([group], time_step, F, CRT, PRT)[0]          # Calculate RR for the group
    group_AR = calculate_AR_group(AT_value, time_step)                         # Use the same AT_value for the entire group
    # Calculate TT_group for the initial component
    TR_group = calculate_TR_group(group_RR, group_AR, time_step)
```

```
    #The actual repair time
    repairtime_group = calculate_Repairtime_group(time_step,PRT,AT_value,[group])[0]

    # Make the gurobi model
    model = gp.Model("time_step_minimization_group")

    # Add variables time_step[t] for component [i]
    time_step_group_vars = model.addVars(len(time_step), lb=0, vtype=GRB.BINARY, name="time_step_group")

    # Add binary variables to indicate if maintenance is scheduled at time step t
    maintenance_scheduled_group = model.addVars(len(time_step), vtype=GRB.BINARY, name="maintenance_scheduled_group")

    # Add objective function to minimize total repair time for component [i]
    model.setObjective(gp.quicksum(TR_group[t] * time_step_group_vars[t] for t in range(len(time_step))), GRB.MINIMIZE)

    # Add constraint sum of time_step[t] for this component == 1
    model.addConstr(time_step_group_vars.sum() == 1, name="time_step_sum_constraint_group")

    # Add constraint to check if maintenance time does not exceed availability
    for t in range(len(time_step)):
        # maintenance_scheduled[t] is 1 if maintenance is scheduled at time step t
        # Ensure that maintenance_scheduled[t] is greater than or equal to time_step_vars[t]
        model.addConstr(maintenance_scheduled_group[t] >= time_step_group_vars[t], name="maintenance_scheduled_constraint_group")

        # The following constraint checks if maintenance time does not exceed availability
        # Use a big M (M_big) to represent an upper bound on maintenance time
        M_big = max(availability) * 2
        model.addConstr(repairtime_group[t] - availability[t] * maintenance_scheduled_group[t] <= M_big * (1 - time_step_group_vars[t]),
name="maintenance_time_constraint_group")


    # Add contraint for reliability threshold of component[i]
    for t in range(len(time_step)):
        model.addConstr(time_step_group_vars[t] * F[component-1][t] <= reliability[component-1], name="reliability_constraint_group")

    # Optimize the model
    model.optimize()

    # Check if the optimization is successful
    if model.status == GRB.OPTIMAL:
        # Retrieve the optimal variable values for this component
        optimal_group_vars = [time_step_group_vars[t].x for t in range(len(time_step))]
        # Find the index of the optimal variable (time step)
        optimal_time_step_index = optimal_group_vars.index(1)
        # Store the optimal time step for this component
        t_opt_group = time_step[optimal_time_step_index]
        TR_sys = TR_group[t_opt_group-1] + TR[value_2-1][sorted_components[1]-1] + TR[value_3-1][sorted_components[2]-1]
        print("TR_sys", TR_sys)
    else:
            print("Error")
    return TR_sys, t_opt_group

TR_sys, t_opt_group_individual = calculate_optimal_group_individual(time_step,t_opt, F, CRT, PRT, TR, AT_value)


def calculate_optimal_group_limit_gas_spring(time_step, t_opt, F, CRT, PRT, TR, AT_value):
    sorted_t_opt = sorted(t_opt.items(), key=lambda x: x[1])
    sorted_components = [item[0] for item in sorted_t_opt]
    value_1 = sorted_t_opt[0][1]
    index = 0
    j = sorted_components[index + 1]
    j2 = sorted_components[index +2]
    # Calculate TT_group for the new component
    new_group = [j] + [j2]
    print("new_group =",new_group)
    group_AR = calculate_AR_group(AT_value, time_step)
    new_group_RR = calculate_RR_group([new_group], time_step, F, CRT, PRT)[0]
    new_TR_group = calculate_TR_group(new_group_RR, group_AR, time_step)
    # Actual repair time for the existing group and the new group
    repairtime_group_new = calculate_Repairtime_group(time_step,PRT,AT_value,[new_group])[0]

    # Make the gurobi model
    model = gp.Model("time_step_minimization_group_new")

    # Add variables time_step[t] for component [i]
    time_step_group_vars_new = model.addVars(len(time_step), lb=0, vtype=GRB.BINARY, name="time_step_new")

    # Add binary variables to indicate if maintenance is scheduled at time step t
    maintenance_scheduled_group_new = model.addVars(len(time_step), vtype=GRB.BINARY, name="maintenance_scheduled_group_new")

    # Add objective function to minimize total repair time for component [i]
```

```
    model.setObjective(gp.quicksum(new_TR_group[t] * time_step_group_vars_new[t] for t in range(len(time_step))), GRB.MINIMIZE)

    # Add constraint sum of time_step[t] for this component == 1
    model.addConstr(time_step_group_vars_new.sum() == 1, name="time_step_sum_constraint_group_new")

    # Add constraint to check if maintenance time does not exceed availability
    for t in range(len(time_step)):
        # maintenance_scheduled[t] is 1 if maintenance is scheduled at time step t
        # Ensure that maintenance_scheduled[t] is greater than or equal to time_step_vars[t]
        model.addConstr(maintenance_scheduled_group_new[t] >= time_step_group_vars_new[t], name="maintenance_scheduled_constraint_group_new")

        # The following constraint checks if maintenance time does not exceed availability
        # Use a big M (M_big) to represent an upper bound on maintenance time
        M_big = max(availability) * 2
        model.addConstr(repairtime_group_new[t] - availability[t] * maintenance_scheduled_group_new[t] <= M_big * (1 - time_step_group_vars_new[t]),
name="maintenance_time_constraint_group_new")
    # # Add contraint for reliability threshold of component[i]
    for i in new_group:
        for t in range(len(time_step)):
            model.addConstr(time_step_group_vars_new[t] * F[i-1][t] <= reliability[i-1], name=f"reliability_constraint_group_new_{i}")
    # Optimize the model
    model.optimize()

    # Check if the optimization is successful
    if model.status == GRB.OPTIMAL:
        # Retrieve the optimal variable values for this component
        optimal_group_vars_new = [time_step_group_vars_new[t].x for t in range(len(time_step))]
        # Find the index of the optimal variable (time step)
        optimal_time_step_index_new = optimal_group_vars_new.index(1)
        # Store the optimal time step for this component
        t_opt_group_new = time_step[optimal_time_step_index_new]
        TR_sys_new = new_TR_group[t_opt_group_new-1] + TR[value_1-1][sorted_components[0]-1]
        print("TR_sys for group of 2 components is: ", TR_sys_new)
    else:
         print("Error")
    return TR_sys_new, t_opt_group_new, repairtime_group_new, new_TR_group

TR_sys_new, t_opt_lg, repairtime_group_lg, new_TR_group = calculate_optimal_group_limit_gas_spring(time_step, t_opt, F, CRT, PRT, TR, AT_value)
print("Optimal time to do maintenance for the group[limit switch, gas spring] is ", t_opt_lg)
```