Titre: Title:	Autonomous Machine Maintenance and Control in Digital Twin Environment
Auteur: Author:	Hussein Adel Taha Hussein
Date:	2022
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Hussein, H. A. T. (2022). Autonomous Machine Maintenance and Control in Digital Twin Environment [Ph.D. thesis, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/10367/

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Autonomous Machine Maintenance and Control in Digital Twin Environment

HUSSEIN ADEL TAHA HUSSEIN
Département de mathématiques et de génie industriel

Thèse présentée en vue de l'obtention
du diplôme de *Philosophiae Doctor*
Génie industriel

Juin 2022

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

Autonomous Machine Maintenance and Control in Digital Twin Environment

présentée par **Hussein Adel Taha HUSSEIN**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Marek BALAZINSKI, président
Soumaya YACOUT, membre et directrice de recherche
Yasser SHABAN, membre et codirecteur de recherche
Walid JOMAA, membre
Tamer GAYED, membre externe

DEDICATION

To my parents, beloved wife, daughter, and son,
I am indebted to you for your sacrifices, encouragement, and endless love.

ACKNOWLEDGEMENTS

I am grateful to my supervisor Soumaya Yacout. I appreciate her guidance and dedication to completing my Ph.D. and writing this thesis in a proficient and timely manner. She knows how to bring out the best in me since we met for the first time. Whenever I went out of the track, she returned me back with her guidance. She once said, "finding the real problem is half the solution, and you have to dig deeper to find it". I have gained more from her wisdom and professional experience. She is one of the best teachers I have ever met, and I feel honored to be part of her team.

It is with great pleasure that I express my appreciation to Yasser Shaban, my co-supervisor, for his encouragement, support, and suggestions.

As members of my jury, I would also like to express my gratitude to Marek Balazinski, Walid Jomaa, Tamer Gayed, and Serge Prudhomme for their constructive input.

I express my deep gratitude to Hussien Hegab for his insight on the practical aspect of most of the thesis work and his valuable discussion. Thanks to all my colleagues and lab members for their support. In particular, I would like to mention Ahmed Sakr as my co-author for the best graduate paper award at the IEOM, Toronto 2019. They are my invaluable friends.

I am thankful to the E-mobility partners "Solutions Serafin" and "ZF" for their support and technical information that enhanced the completion of this thesis.

I would like to acknowledge Mitacs Canada for their grant Research-Training-Award 2020 and their funding for the majority of this research.

RÉSUMÉ

Cette thèse présente le développement de systèmes intelligents dans l'environnement du jumeau numérique pour prolonger la disponibilité du système industriel et maintenir les performances système souhaitées. Des algorithmes basés sur l'apprentissage automatique (ML) ont été développés pour surveiller et détecter les performances de dégradation des systèmes physiques en ligne tout en analysant les données des capteurs du système. D'autres algorithmes basés sur ML sont développés pour définir les actions capables de restaurer de manière autonome les performances du système au niveau souhaité. Ces algorithmes ont été développés pour fonctionner en ligne dans l'environnement du jumeau numérique et pour déclencher et activer le module d'action autonome. Le module autonome coopère avec le module de surveillance de la santé et l'action corrective est ajustée au système physique pour formuler l'autonomie en boucle fermée.

Les systèmes de maintenance autonomes et numériques développés ont été mis en œuvre dans différents systèmes industriels. Ces approches traitent de l'auto-guérison pour l'extension du temps de disponibilité après une panne et avant une panne. Les algorithmes d'apprentissage automatique sont appliqués à différents types de données, telles que des séries chronologiques et des données numériques indépendantes du temps. La reconnaissance de modèles par analyse logique des données (LAD) active le mécanisme d'auto-guérison et définit la cause de l'échec qui doit être atténuée. L'intégration de Deep Reinforcement Learning (DRL) et de LAD se traduit par un module autonome pré-défaillance qui ralentit la dégradation des actifs et augmente le temps de défaillance (T2F). La combinaison d'actions autonomes et de principes de maintenance centrée sur la fiabilité (RCM) permet les actions de maintenance numérique sur des systèmes dont les données de défaillance opérationnelle sont incertaines ou indisponibles. Le déploiement des approches développées impacte les performances du système et améliore sa disponibilité. Le mécanisme d'auto-guérison développé ralentit la dégradation et prolonge le temps jusqu'à la défaillance d'un système. Les approches pré-panne prolongent le temps de fonctionnement et e temps de défaillance (T2F) en moyenne de 40% dans les systèmes industriels et de 35% dans les applications Electric-Bus.

ABSTRACT

This thesis presents the development of intelligent systems in digital twin environment to extend the industrial system's uptime and to maintain the desired system performance. Machine learning (ML) based algorithms were developed to monitor and detect the degradation performance of the physical systems online while analyzing the system's sensor data. Other ML-based algorithms are developed to define the actions that are able to autonomously recover the system's performance to the desired level. These algorithms were developed to run online in digital twin environment, and to trigger and enable the autonomous action module. The autonomous module cooperates with the health monitoring module and the corrective action is adjusted to the physical system to formulate the closed-loop autonomy.

The developed autonomous and digital maintenance systems were implemented in different industrial systems. These approaches address the self-healing for post-failure and for pre-failure uptime extension. The machine learning algorithms are applied to different types of data, such as time-series and numerical time-independent data. Logical Analysis of Data (LAD) pattern recognition enables the self-healing mechanism and defines the cause of failure that is to be mitigated. The integration of Deep Reinforcement Learning (DRL) and LAD results in a pre-failure autonomous module that decelerates the asset degradation and increases the time to failure (T2F). Combining autonomous actions and Reliability Centered Maintenance (RCM) principals allows the digital maintenance actions on systems that have uncertain or unavailable operational failure data. The deployment of the developed approaches impacts the performance of the system and improves its uptime. The developed self-healing mechanism slows down the degradation and extends the time to a system's failure. The pre-failure approaches extend the uptime and the Time to Failure (T2F) on average by 40% in industrial systems and 35% in Electric-Bus applications.

Keywords: Maintenance 4.0; Autonomous Machines; Uptime; Machine learning; Artificial Intelligence; Digital Twin.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

AI	Artificial Intelligent
CBM	Condition-Based Maintenance
CPS	Cyber Physical System
DLSTM	Deep Long Short Term Memory
DoF	Degree of Freedom
DRL	Deep Reinforcement Learning
DT	Digital Twin
EIS	Engineering Immune System
E-Mobility	Electrified Mobility
E-Bus	Electric Bus
FCN	Full Connected Neural Network
FTC	Fault Tolerant Control
LAD	Logical Analysis of Data
MAE	Mean Absolute Error
MTBF	Mean Time Between Failure
MILP	Mixed Integer Linear Programming
NN	Neural Network
P2P	point to point
P2D	Point to Distribution
PBAF	Predicted Before Actual Failure
PHA	Position Health Assessments
PHM	Proportional Hazard Model
RAM	Regression Adjustment Multivariate
RMSE	Root Mean Square Error
RUL	Remaining Useful Life
STFT	Short-time Fourier Transform
TCP	Tool Center Point
TCP_d	Tool Center Point displacement
T2F	Time to Failure
W.M.E-Bus	Wheel Motor Electric Bus

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

It is crucial to maintain the uptime of machines. As technology is advancing more quickly than ever, uptime improvement activities have become an urgent matter. With the Industry 4.0 revolution, systems are fully automated and connected to digital platforms. Therefore, more variabilities are induced to physical machines, which could impact the reliability of the machines. Additionally, it is difficult to conduct physical maintenance or repairs immediately due to a lack of human resources. Thus, industrial machines need to be upgraded and digitized to become autonomously maintained machines that could automatically extend their performance and uptime.

1.1 Frontier of Knowledge

This section presents the basic concepts that support the development of the proposed autonomous and self-healing machines. At the beginning, the terms for some basic concepts are stated as follows:

- **Robust system:** a system that is capable of performing its intended function even in the presence of noise. When its robustness continues for a specific amount of time, the system is called a reliable system [5, 6].
- **Reliability:** The probability that the system will survive under given conditions without failing for a specific period of time [5, 7].
- **Anomaly:** An abnormal, irregular or inconsistent situation [8].
- **Fault:** A defect or imperfection that develops when physical degradation has occurred, but this degradation is not severe enough to be termed as a failure [8].
- **Failure:** The event that occurs when a machine is not able to perform its required function [8].
- **Breakdown:** A specific type of failure in which the machine or system's component is completely unable to function [8].
- **Uptime:** The time that a machine/system is in service and can perform its desired function [8].

- **Stable system:** The system that has an output response, which lies in an acceptable boundary [9].
- **Fault tolerance:** The ability of a system to perform its function despite the presence of a fault [6]. A popular fault tolerance mechanism is the redundancy of components [8].
- **Self-awareness system:** The system that is capable of monitoring its own degradation and detecting its own anomalies [10].
- **Self-healing system:** A system that detects the causes of failure and that performs a recovery action to return the system itself to the state that confirms its function, while taking into consideration its degradation [5, 11].
- **Autonomous system:** A system that is capable of managing its state without depending on external factors, an autonomic control finds a closed loop relation between the sensor data collection and the system's behavior [5].
- **Digital Twin (DT):** The model that emulates the physical machine in the cyber/digital environment and has the capability of interacting with the real machine in the physical environment [12–14].

By definition of self-healing in autonomic systems, it is capable of performing two main functions, namely: (1) the detection of anomalies or failures, and (2) executing the recovery actions that eliminate the cause of failure. These two functions are interconnected to close the autonomic loop and clear a failure without disrupting the system.

1.1.1 Health Monitoring and Diagnosis

The health monitoring module is one of the limbs of an autonomously maintained machine. This module detects and diagnoses the faults, while analyzing the machine's sensor data [5, 15, 16]. By deployment of the health monitoring module, the machine becomes a self-awareness system [10, 15]. It reads the online sensor data at each time t, and analyzes these data to predict the instant of fault or failure and the Remaining-Useful-Life (RUL) of that machine [16–18]. Once, the fault is detected, this module triggers the autonomous action module [5, 10].

1.1.2 Self-healing and Autonomous Action

This is a machine-learning based module that analyzes the state of the machine and defines the proper recovery action to take [16, 17]. The autonomous action module interacts with

the physical machine and provides the automatic corrective actions to improve the machine's uptime [15, 17]. The performed action at time t is evaluated according to the updated sensor data of the machine at time t+1. The autonomous actions include three types of actions: Pre-Failure, Post-failure, and triggering for spare replacement. Both modules of the autonomously maintained machine are developed in a Digital Twin environment [17]. Figure 1.1 depicts the integration diagram of the two autonomous machine's modules and the physical machine.

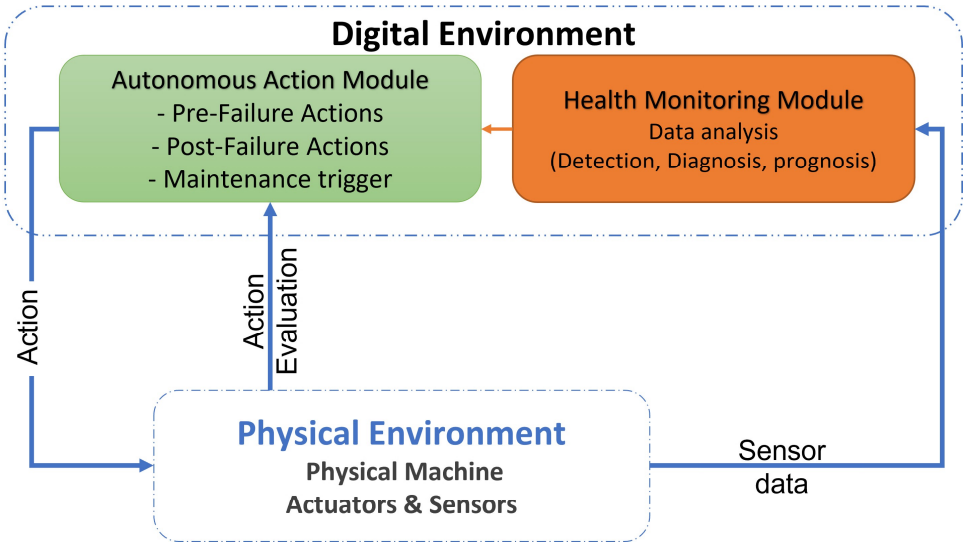


Figure 1.1 Diagram of Autonomous Maintained Machine Integration

1.2 Problem statement

According to the topics discussed in previous sections, the problem statement is formalized and the following research questions will guide the research in this thesis:

- How can unseen/unexpected anomalies be detected, identified, and categorized?
- How quickly and accurately will the proposed approaches deal with fault detection and recover the system before its breakdown?
- How can a self-healing mechanism be implemented? How could it be linked to the failure cause?
- To what level can the self-healing approaches improve the operational performance of the system?

- How could the unavailability of data impact the implementation of autonomous machines?

1.3 Research Objectives

The main objective of this research is to develop different approaches for self-healing and autonomous uptime-improvement for the operating machines. The proposed machine-learning and data-driven approaches integrate the capabilities of Digital Twin and Artificial Intelligence (AI) to perform autonomous actions that mitigate systems in the pre-failure and post-failure states. Online and real-time data is analyzed to predict and detect the anomalies and performance degradation. The root cause analysis is performed to develop the corrective actions that are executed through an autonomic closed-loop. This main objective is divided into four sub-objectives as follows:

- **Objective 1: Functional Failure and Anomaly Detection, Diagnosis, and Prognosis.**
  This objective address the modeling and simulation of a diagnosis and prognosis health assessment technique. The health assessment tool detects machine anomalies due to normal and abnormal events such as normal degradation of electrical and mechanical systems, and abnormal events such as anomalies and faulty events. This objective includes developing a machine learning tool that can be integrated into the DT online to monitor performance degradation, which can be stopped and redressed if proper action has proceeded.

- **Objective 2: Develop a Self-healing Mechanism for Post-failures**
  This develops a DT self-healing mechanism that analyzes the controllable variables of the machine to perform certain recovery actions autonomously. Modeling and simulation of a proactive ML algorithm to be interconnected with the health monitoring and detection module (objective 1) to provide online self-healing actions that are quick enough to recover the machine status in post-failure status. Development of self-healing mechanism that is easy to be integrated to the operating machines without additional hardware requirements.

- **Objective 3: Develop Pre-Failure approach to improve the systems performance**
  Objective 3 involves developing an approach to monitor system performance's degradation in Pre-Failure stage and continuously executes autonomous-actions that improve

system performance. Machine-learning algorithms are developed to monitor the performance degradation and define the penitential failure point of the system studied. This approach addresses the best actions to take, considering Time to Failure (T2F) extension, productivity, and smoothing the transition to slowdown the degradation rate of the systems.

- **Objective 4: Autonomous uptime-improvement Scheme**
  It Develops a general scheme to link failure analysis and autonomic actions to improve the uptime of a system with data unavailability. This scheme models a closed-loop relationship between the uptime of the system and its online operating conditions. The scheme that is developed combines the system failure modes, symptoms, causes, and different healing modes. It includes recommended autonomous actions to improve the uptime of the systems studied.

1.4 Originality of The Research

This research develops novel approaches to enable autonomous maintained machines. To the best of our knowledge, the following topics have not been discussed in the literature.

- Developing a novel self-healing mechanism based on the LAD Machine learning algorithm to detect and define the failure root cause. This mechanism enables online self-healing with interpretable fault-recovery actions.
- Developing a model-free approach that addresses the system performance and extends its Time-to-Failure (T2F) in the pre-failure interval. A combination of Reinforcement Learning (RL) and the LAD algorithm detects the potential failure point of the system and executes continuous actions to increase the T2F and to keep the system operating at an acceptable performance level.
- Developing an autonomous scheme to improve the uptime for electric buses with data unavailability or data uncertainty. Extension of the Reliability Center Maintenance (RCM) concept to include autonomous improvement actions. Adapting the Fuzzy-FTA algorithm to analyze the failure of electric buses and to define the bus's components that have the potential to fail.
- Developing an approach that combines the Long-Short-Term-Memory (LSTM) and Regression Adjustment for Multivariate (RAM). This approach enables the monitoring and detection of anomalies for a robotic arm while the data variables are correlated and unsupervised.

- Developing a machine learning-based framework for the implementation of maintenance 4.0 in the aerospace industry. This framework addresses the cost-reduction of system implementation and decreases the number of required sensors.

These are general points about the originality of this study: more detailed ones are stated in each of the following chapters.

CHAPTER 2 THESIS ORGANIZATION

2.1 Organization of the Thesis

Figure.2.1 illustrates the structure of the thesis to implement the autonomous up-time enhancement in industrial systems according to operational data. This thesis is divided into eight chapters, as follows:

- **Chapter One** introduces the research problem and the general objectives.
- **Chapter Two** reviews the critical Literature and related research.
- **Chapter Three** is the current chapter and it shows the thesis organization and research deliverable.
- **Chapter Four** addresses the detection of anomalies.
- **Chapter Five** develops a self-healing mechanism to recover the machine after the incident of the fault.
- **Chapter Six** is a maintenance 4.0 framework, and it addresses the prediction of the Remaining Useful Life (RUL) for aircraft engines. Different machine learning algorithms are developed to analyze the system run-to-failure data.
- **Chapter Seven** is a Pre-Failure approach to slowdown a system's degradation and increase the T2F. The developed approach is verified on a CNC machine.
- **Chapter Eight** studies a failure analysis for complex integrated systems that does not have available operational data. The system is dismantled to its basic components to defend the failure causes and system's critical components. The proposed approach is implemented in a wheel Motor Electric-Bus (W.M.E-Bus).
- **Chapter Nine** is a general scheme to develop and implement the autonomous corrective approaches in the Digital Twin environment. This scheme tackles the unavailability of operational data or data that contains uncertainty. Its objective is to improve the uptime of complex integrated systems that have critical failure consequences. An autonomic closed loop is developed to execute uptime improvement actions while monitoring real operating conditions. The proposed scheme is verified on a W.M.E-Bus to increase its uptime.

- **Chapter Ten** provides a general discussion.
- **Chapter Eleven** is the conclusion and future work.

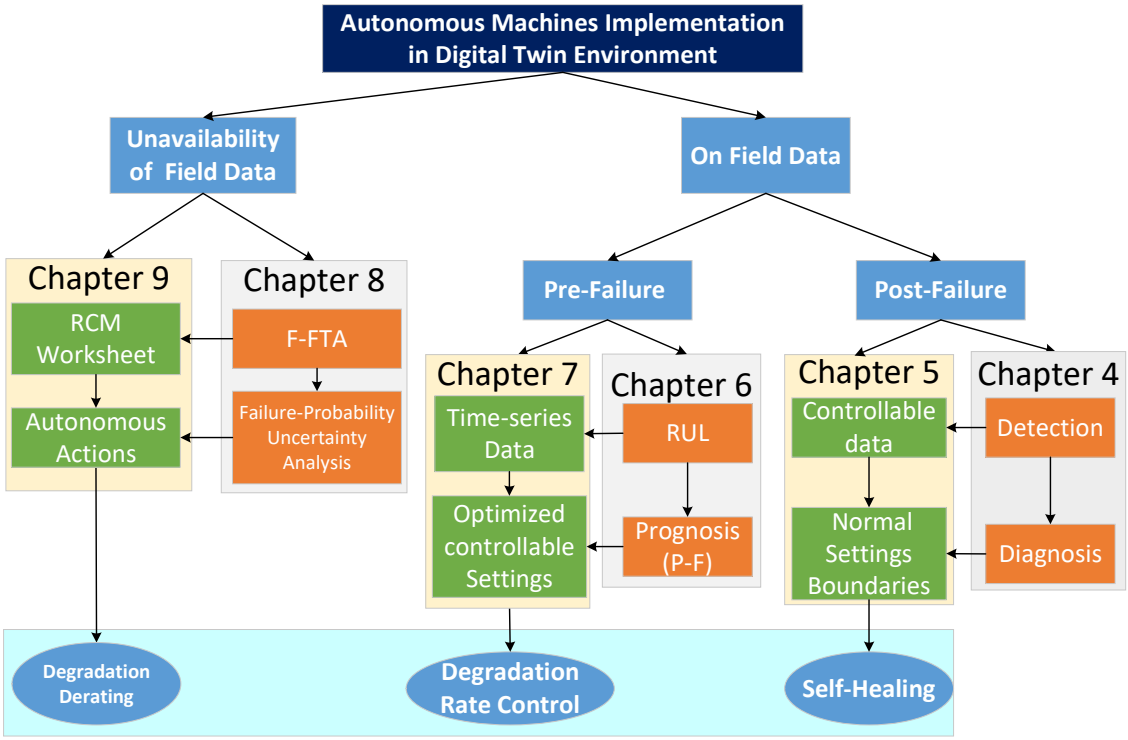


Figure 2.1 Thesis structure

2.2 Research Deliverable

The outcomes of this research are six articles that cover the four main objectives provided in section 1.3, as follows:

1. **Detection and Monitoring for Anomalies and the Degradation of a Robotic Arm Using Machine Learning**
    - **Published in:** Advances in Automotive Production Technology–Theory and Application. Springer Vieweg, Berlin, Heidelberg,2021.
    - **Abstract:** Robotic arm performance varies due to normal and abnormal events. Normal events may include degradation of equipment, motors, mechanical system joints, and gears, while abnormal events may occur such as faulty episodes. In this

paper, we address positional performance degradation that can be stopped and redressed if suitable required action is achieved. The Tool Center Point (TCP) position measurement devices are expensive, hence unavailable to every robot. Some industrial processes are critically sensitive to target tool position such as assembly, pin and past, and material handling. We propose a data driven artificial intelligence tool to detect anomalies and degradation of the robotic arm for a positional health assessment without the need for special advanced sensors. TCP deviation is predicted using deep machine learning models that train on a time series of historical data of the robot's performance. Statistical thresholds are calculated to detect the robotic arm's degradation and anomalies by performing residual analysis. An alarm system is built by applying the proposed monitoring tool online.

2. **Autonomous self-healing mechanism for a CNC milling machine based on pattern recognition**

   - **Published in:** Journal of Intelligent Manufacturing.
   - **Abstract:** A sustainable and reliable machining process is the main goal of seeking machine digitization. Artificial Intelligence (AI), and Cyber-Physical System (CPS) combined with Artificial Intelligence are used for process control. This has become more essential in the case of machining of high-cost aerospace materials and critical product specifications. In this paper, a novel self-healing mechanism was developed to recover a CNC machine from producing parts that do not conform to surface roughness's specifications. The machine settings are reconfigured autonomously and online to recover from the effect of tool wear and to keep the surface roughness within the design specifications. The proposed self-healing mechanism is based on a pattern recognition algorithm called Logical Analysis of Data (LAD). This algorithm generates patterns that characterize the out-of-specification state, and provides a corrective setting within the recovery patterns of the within-specification state by using various distance approaches. The developed self-healing mechanism is composed of three modules: CPS model of the CNC machine (module 1), classification into, out of, or within-specification states (module 2), and a self-healing controller (module 3) that is activated if the state of out-of-specification is found by module 2. The three modules are software. The current hardware system of the machine is not altered. The proposed self-healing mechanism is validated on CNC machines with a wide range of machining param-

eters of feed rate from 20 mm/min to 700 mm/min and spindle speed from 10,000 RPM to 40,000 RPM. To validate the developed mechanism, a deep learning artificial model was developed on physical data to emulate the CNC milling machine in a CPS simulation environment, and test runs were executed. The proposed self-healing mechanism was evaluated under several simulation runs that covered the ranges of CNC machine settings. The measure of performance of the proposed mechanism is the out-of-specification clearing time. The validation runs show that the proposed self-healing mechanism was able to clear the out-of-specification state and to recover the within-specification state in less than three seconds, with the best distance metric approach. The results of the time response for each test run are reported.

3. **Aircraft Engine Remaining Useful Life Prediction Framework for Industry 4.0**

   - Best graduate paper award, IEOM, Toronto 2019
   - **Abstract:** This article proposes a Condition-Based Maintenance (CBM) approach for aircraft engines and Remaining Useful Life (RUL) monitoring, and failure prevention. Due to the unavailability of run-to-failure data, Turbofan Engine Simulation data, obtained from NASA repository, is used to train and test our model. Data Acquisition and Management system framework and planning are proposed for online monitoring and RUL prediction. In practice, sensor measurements usually suffer from noise contamination, hence the prediction models are challenged by noise contaminated data for both training and testing tasks. This is done to assess their prediction ability in a similar condition of having noisy data. Linear and nonlinear prediction models are developed, with performance comparison addressing both regression and classification problems. Models performance indices consider both prediction accuracy and percentage of predictions before the actual failure (PBAF). The proposed model considers continuous learning and improvement to account for any further operational changes that affect the model prediction ability. This is reached by ingesting the model with the actual RUL during the maintenance of the engine unit, and by comparing it to the predicted one.

4. **Deep Reinforcement Learning for Autonomous Pre-Failure Tool-Life Improvement**

- **Accepted with minor reversion in:** International Journal of Advanced Manufacturing Technology.
- **Abstract:** This paper develops an approach to improve a CNC machine's tool performance and slow down its degradation rate automatically in the Pre-Failure stage. A Deep Reinforcement Learning (DRL) agent is developed to optimize the machining process performance online during the Pre-Failure interval of the tool's life. The Pre-Failure agent that is presented in the proposed approach tunes the feed rate according to the optimal policy that is learned in order to slow down the tool's degradation rate, while maintaining an acceptable Material Removal Rate (MRR) level. The machine learning techniques and pattern recognitions are implemented to monitor and detect the tool's potential failure level. The proposed mechanism is applied to a CNC machine when turning Titanium Metal Matrix Composites (TiMMC). A CNC machine Digital Twin (DT) is developed to emulate the physical machine in the digital environment. It is validated with the physical machine's measurements. The proposed pre-failure mechanism is a model-free approach, which can be implemented in any machining process with fewer online computational efforts. It also covers a wide range of cutting speeds, up to 15,000 RPM. Deployment of the proposed machine learning approach for the particular case study improves the tool's Time to Failure (T2F) by 40% and the MRR by 6%, on average, compared to the classical approach.

5. **Failure Reasoning and Uncertainty Analysis for Wheel Motor Electric Bus**

- **published in:** 26^{th} IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, 2021.
- **Abstract:** Wheel Motor Electric Bus (W.M.E-Bus) is a recent e-mobility technology, which has a complex system integration. Since the operational reliability and life cycle data of such systems is scarce, it becomes impractical to plan for maintenance and determine system-critical components. Moreover, E-Bus system dismantling and assembling is a long time process especially for components near to the its Power-system. In this paper, we propose a Fuzzy-logic fault-tree evaluation for the W.M.E-Bus system under uncertain failure data. The proposed method indicates the critical components that significantly influence the system's failure uncertainty. At 10% failure rate uncertainty, control unit failure, including the embedded software, is ranked the top critical failure mode with 1.8 Fuzzy Importance Measure (FIM).

6. **Autonomous Uptime-Improvement for Electric-Bus in Digital Twin Environment**

   - **Submitted to:** IEEE Transactions on Intelligent Transportation Systems
   - **Abstract:** Autonomous and self-improvements systems rely on the historical operational data of these systems. With the lake of operational data, the development of the self-improvement process becomes more challenging. This paper develops an autonomous uptime improvement scheme for a system the unavailability or uncertainty of the operational failure data. Fuzzy-Fault Tree Analysis (F-FTA) is adapted to analyze the system failure with operational data uncertainty. Meanwhile, the Reliability Centered Maintenance (RCM) worksheets are developed to define the uptime-improvement autonomous actions. The proposed scheme is verified on the Wheel Motor Electric Bus (W.M.E-Bus) drive system in Digital Twin (DT) environment. It links the on-road Physical W.M.E-Bus driving condition to the system failure and reliability estimation, which are in the digital DT environment. The Embedded Control Unit (ECU) transmits the W.M.E-Bus's driving cycle to be analyzed in the digital environment. Then, the improvement action of derating percentage for the driving cycle is sent back to the E-bus to automatically manipulate the driver's pedal in the next driving cycle. In this particular case study, the deployment of the proposed scheme increases the maximum allowable W.M.E-Bus millage distance by 8603 miles when it reduces the driving cycle by 36 % on average. A detailed improvement analysis for different operating speeds and RCM worksheets are reported.

CHAPTER 3 CRITICAL LITERATURE REVIEW

3.1 Fault-Tolerant Control Systems

Fault-tolerant control (FTC) is a set of recent techniques that were developed to increase plant availability and reduce the risk of safety hazards. It aims to compensate for fault effects on the system during operation to maintain system stability regardless of the nature of the fault. Fault-tolerant control merges several disciplines to achieve this goal, including online fault detection, automatic condition assessment, and calculation of remedial actions when a fault is detected [6, 9]. In FTC, a fault is defined as a deviation in the parameters of the differential equations that modeled the system. Fault tolerance is defined as the ability of the system to continue its function regardless of its faults [6, 9]. Passive FTC (PFTC) is considered the traditional way of FTC, which is limited to a few faults that were defined in the design phase of the machine. On the other hand, Active FTC (AFTC) consists of a Fault Detection and Isolation (FDI) module in addition to a reconfiguration mechanism [6, 9, 19]. AFTC has a more complex architecture, with a slower response time than PFTC. However, it can accommodate various fault types. It is used to design an online controller to tolerate faults in sensors, actuators, or system disturbances, and to achieve the stability of the system [6,6,19]. The AFTC system is divided into four sub-systems: (1) a reconfiguration controller, (2) an FDI module, and (3) a controller reconfiguration mechanism. The controller mechanism selects the configured controllers, 1 to N, according to the FDI signal. Each controller is previously configured to cover certain system stability rang under a certain FDI signal condition. As system complexity and nonlinearity increase, Lyapunov equations will become more complex to model, and they will need more computation time, as the reconfiguration mechanism's response time increases. AFTC takes a large computation time online for fault estimation and reconfiguration mechanism. Moreover, AFTC is a complex architecture [6, 9]. The reconfigured controller is built on traditional control theories and nonlinear mathematical models. This means that the AFTC reconfiguration mechanism does not learn from previous experience. Consequently, for the detection of new/undersigned faults, they produce erroneous decisions [6, 19]. Also, for the FDI module, the accuracy with a neural network is not guaranteed [20, 21]. Finally, fault-tolerant control is considered a reactive system that handles post-failures and does not address a machine's degradation, which needs an active system to consider the pre-failure status [5]. PFTC is limited to several faults that have been defined in the design phase. If the incident fault is not included in the previously designed ones, this may lead to a system breakdown. Both types of FTC have

several industrial limitations. FTC is considered to be a reactive technique because it reacts after the incident of failures [5].

3.2 Maintenance in Cyber-Physical Systems (CPS)

Maintenance 3.0 is aimed at reaching high equipment availability by avoiding the failure occurrence. In that sense, it is a pre-failure action. It is based on condition-based maintenance [5,22]. In Industry 4.0, the IIOT technologies lead to the development of cyber-physical platforms, where equipment and devices are fully connected to a central network [16,23]. This means that all of the machines are aware of their status as well as the status of other machines in the production line [16, 22, 23]. As such, more information is available online and in real-time, and maintenance actions can be updated instantaneously to avoid failure. J. Lee, et.al studied the Prognosis Health Management (PHM) of a saw-cutting machine that implemented the five levels of CPS architecture [10,24,25]. This study developed a prognostic tool for self-awareness. This tool detects anomalies due to saw blade degradation. The study highlighted diagnosis and prognosis in the CPS environment. CPS research aims to have full integration of the five levels of network control, data analysis, learning theories, simulation, and visualization. It enhances the incorporation of engineering disciplines to achieve a fully autonomous and efficient process [25]. In this thesis, we consider the digital environment of the CPS to have a self-healing and auto-maintained machine.

3.3 Reliability Centered Maintenance (RCM)

Reliability and maintenance are based on the efficient definition and diagnosis of anomalies and failures [26–28]. They are also based on finding the optimal actions of maintenance or replacement time [29]. Reliability Centered Maintenance (RCM) is a concept that was introduced to reduce failures caused by inadequate maintenance for several industries [30]. Maintenance is applied to ensure that the machine continues to perform as intended, while RCM addresses modes of failure and their consequences and possible maintenance actions. These actions are chosen to improve the maintenance function and minimize the influence of failures because of inadequate decisions. The RCM approach identifies the potential causes of failure, failure effects or consequences, and possible actions to prevent or reduce the risk of failure [30]. Effects of machine failure and failure consequences consider human, environmental, and operational risks. To prevent failure, the failure mechanism of each failure mode is identified. One of the most common tools to clarify and understand failure causes is a Fault Tree Analysis (FTA). FTA is a top-down approach that starts with anomalies at a

top-level, then moves down until reaching the root causes or basic events level. In the middle levels, there are logical combinations of intermediate events that lead to anomalies [29]. A graphical representation of FTA leads to failure mode mechanism. Hence, the recommended RCM actions become justifiable. It also shows the shortest way to prevent failure. In this research, RCM concepts are used to reach objective 4.

3.4 Logical Analysis of Data (LAD)

LAD is a non-statistical supervised data mining method. LAD combines Boolean logic functions and combinatorial optimization. It is capable of solving classification problems [31]. The idea of LAD was introduced by Peter L. Hammer in [20]. The advantage of LAD over other classification methods is that it generates interpreted logical patterns for each fault [20, 21]. LAD has been used successfully in condition maintenance [20, 21, 31]. It shows excellent performance in knowledge extraction for supervised and semi-supervised learning problems. The cbmLAD software was used for condition-based maintenance by Yacout et al. [32]. cbmLAD is also used in several applications and multiclass industrial fault diagnosis and prognosis problems [20,21,31]. The main cbmLAD processes are (1) Data binarization for labeled data, (2) pattern generation, and (3) classification with discriminant functions [32]. CbmLAD generates understandable and more interpretable and strong classifying patterns. In this research, cbmLAD is used to obtain the interpretation patterns for the controllable and uncontrollable variables of the machine. LAD does not require data pre-processing or any statistical assumptions, which make it a general applicable tool for any kind of machine or data. Moreover, cbmLAD does not require high computation, which makes it a robust applicant to achieve objective 2 and 3.

3.5 Reinforcement Learning for Continuous Control

The standard RL consists of a decision taken in a discrete time step. At time step t, a virtual agent receives state S_t, takes action a_t and finds the reward r [33, 34]. The RL goal is to find the policy π that maximizes the expected state return J. Q-learning is an off-policy algorithm that uses a greedy policy. The selected action is what maximizes the returned Q-value. It is impossible to apply Q-learning to a continuous action space environment, where to find the greedy policy, it is required to optimize action a_t at each time step, which is difficult. The actor-critic approach is used to solve this issue with the Deterministic Policy Gradient (DPG) algorithm [34]. The Critic is an action-value function used to calculate the temporal difference (TD) error to criticize actions made. The actor

is deterministic policy function used to choose action a_t given state S and it is updated according to expected return J [33, 34]. A Deep Deterministic policy gradient (DDPG) algorithm is an implemented deep Q-network on DPG to learn in large state and continuous space. In off-policy algorithms, the exploration is independent of the learning process [35]. DDPG challenges the exploration policy μ' with random noise [34]. DDPG was implemented for set point tracking control of a complex chemical process with non-linearity and noise [33]. The implemented DDPG consisted of a double deep network to consider continuous action needs. The environment was represented by mathematical differential equations. The reward function is calculated as the difference between a set point and sensor measurement. DDPG was developed as a toolbox for the speed control of various types of electrical motors [36]. The main items in an electrical rotating machine environment are an electric motor, motor drive, and mechanical load. Electrical motors are categorized according to the type of power supply, such as Alternating Current (AC) or Direct Current (DC) and theory of operations. Each motor has a specific design of a motor's drive. A.Traue, et.al (2019) implement the DDPG to control the speed of DC and Permanent Magnet Synchronous motors environments, including the motor drive with variable mechanical loads [36]. The main feature of DDPG computing is its highest performance, with a continuous control environment [33,34]. DDPG is a model-free, continuous environment, off-policy algorithm that uses a replay buffer that is stored and updates states, actions and rewards for previous time steps during training [37]. In this research, DDPG is used to achieve objective 3 of optimal proactive self-healing action. It learns and updates the algorithm, and provides a continuous action space [33, 34].

CHAPTER 4 ARTICLE 1: DETECTION AND MONITORING FOR ANOMALIES AND THE DEGRADATION OF A ROBOTIC ARM USING MACHINE LEARNING

Hussein A. Taha, Soumaya Yacout, Lionel Birglen
Published in: Advances in Automotive Production Technology–Theory and Application.
Springer Vieweg, Berlin, Heidelberg,2021.
Doi:https://doi.org/10.1007/978-3-662-62962-8_27

Abstract
Robotic arm performance varies due to normal and abnormal events. Normal events may include degradation of equipment, motors, mechanical system joints, and gears, while abnormal events may occur such as faulty episodes. In this paper, we address positional performance degradation that can be stopped and redressed if suitable required action is achieved. The Tool Center Point (TCP) position measurement devices are expensive, hence unavailable to every robot. Some industrial processes are critically sensitive to target tool position such as assembly, pin and past, and material handling. We propose a data driven artificial intelligence tool to detect anomalies and degradation of the robotic arm for a positional health assessment without the need for special advanced sensors. TCP deviation is predicted using deep machine learning models that train on a time series of historical data of the robot's performance. Statistical thresholds are calculated to detect the robotic arm's degradation and anomalies by performing residual analysis. An alarm system is built by applying the proposed monitoring tool online.

Keywords: Robotic Arm, Position Health Assessment, residual analysis, machine learning.

4.1 Introduction

Industrial robots represent an important part of industrial manufacturing and ma-chining process. They are critical to industrial processes, such as assembling, and are sensitive to robotic arm tool displacement. A small displacement or deviation leads to line stop, product reworking, and process down time. Robotic arms are challenged with significant variability and positional uncertainties that frequently result in robot position failure. Malfunctioning robots cost industrial plants more than 20K$/min [38]. Robotic arm position health assessments have been addressed in the literature to solve displacement problems and enhance

overall accuracy [38–41]. However, this requires highly accurate advanced measurements that are difficult to be found in each manufacturing plant. Meanwhile, it is not easy to send these robots to be calibrated every working day.

In recent years, researchers have worked to address fault detection of industrial robot arms. Their work is categorized into two groups, a model-based [38,40,42,43] and historically-based model [44, 45]. The model-based methods use mathematical equations to model the physics of the robotic arm and errors in the inverse kinematic matrices. Manish Goel, et.al [40] used a model-based method to study unified locked-joint failure. These authors considered one type of manipulator fault of a locked joint. The robot arm was analyzed under a faulty state over all feasible space the robot could reach. This method was used to model 2 Degrees of Freedom (Dof) and 3 DoF simple robotic arms. However, this method needs significant computational power, thus it is difficult to be applied to the online operation of a 6 DoF robot arm. Guixiu Qiao, et.al 2017 [42] developed a mathematical model to detect the positional error using an inverse kinematics model for a 6 DoF robot. This method has an industrial limitation, since it requires the measurement and the recording of the robot feasible working space using advanced 7-D measurements before operation. Yizheng Zhang, et.al 2019 [38] used QR codes and an advanced camera to detect robotic arm joints that have had a failure for pick-and-place experiments. The proposed method needs the presence of cameras, hence a clean environment to obtain precise measurements.

The other detection methods are based on analyzing the robotic arm's stored data. Costa, Marcelo, et.al 2019 [44] developed a hybridization of a boosting classifier to detect joint failure for a 6 Dof industrial robot. They assumed that the data fit a lo-gistic distribution. This proposed method was a failure detection approach, and it did not address the robot degradation anomalies. Qibo. Y, et.al 2020 [45], studied the fault diagnosis for ball screw industrial robots. The current fault's signature was characterized to diagnose two types of faults. The developed diagnostic model used short-time Fourier trans-form (STFT) signal analysis and logistic regression classifiers to detect a faulty state. STFT signal analysis outperforms the wavelet decomposition. Current ball screw data consists of 20 samples for each fault and 20 samples for motor health status.

The main contribution points of this paper are: (1) the development of an online data-driven positional health assessment tool for robotic arms. This tool monitors and detects displacement anomalies regardless of measurements' correlations. (2) The development of a time series Artificial Intelligence (AI) model to predict robotic arm displacements before incident or failures. (3) Cost savings, since the developed tool does not need special advanced displacement measurements in a robot's operational environment.

This paper is organized as follows: Section 2 contains the robotic arm data description. Sec-

tion 3 is a summary of the proposed methodology. Section 4 presents the obtained results, and the conclusion is summarized in Section 5.

4.2 Data Description

The proposed data-based method ingests a time-series data for a six degree of freedom robotic arm with six joints. The robot has a payload of 5 kg. The speed of all joints is 180 o/S, the tool speed is 1 m/s, and the repeatability is ± 0.1 mm. Table 4.1 shows the description of input and response variables that were used for the proposed supervised learning model. This robotic arm data is from laboratory data and was acquired by the National Institute of Standards and Technology's (NIST) with a frequency of 125 Hz for six position episodes [40]. The data response variables are the Tool Center Point (TCP) Cartesian coordinates (x, y, and z) and it was measured with laser sensor that has an index of [55–57] in Table 4.1.

Table 4.1 Descriptions of data

Variable Index	**Variable name**	**Variable type**
1–6	Target joint [1–6] positions (angle)	Setting
7–12	Actual joint [1–6] positions (angle)	Measurement
13–18	Target joint [1–6] velocities (m/s)	Setting
19–24	Actual joint [1–6] velocities (m/s)	Measurement
25–30	Target joint [1–6] currents (m/s)	Setting
31–36	Actual joint [1–6] currents (A)	Measurement
37–42	Target joint [1–6] accelerations (m/s2)	Setting
43–48	Target joint [1–6] torques	Setting
49–54	Joint control [1–6] currents (A)	Measurement
55–57	Actual Cartesian coordinates (x, y, z) of the TCP (m)	Response

4.3 Methodology

4.3.1 TCP displacement prediction

We propose a deep Long Short-Term Memory (DLSTM) model to predict robotic arm tool displacement, since the robot data is a time series. Our proposed DLSTM model consists of two main Long Short-Term Memory (LSTM1, LSTM2) layers, two drop-out layers, and a Full Connected dense Neural network (FCN) layer. The DLSTM prediction model architecture is shown in figure 4.1. Its input is the robotic arm's joint data variables that have an index of $[1:54]$ in Table 4.1 with time window 400 m sec (50 samples). Meanwhile, the output is the predicted TCP displacement for the 51st sample in the next time window. It is the

Euclidean distance given in equation 7.1 and it depends on the response variables that were indicated in Table 4.1.

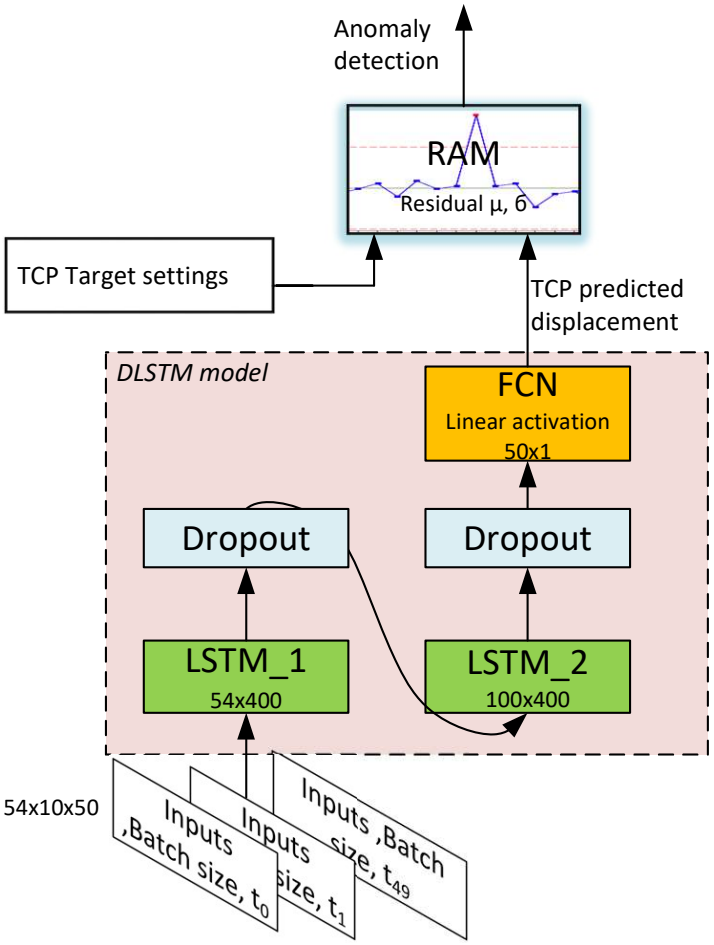


Figure 4.1 Schematic of the proposed DLSTM-RAM robotic arm anomalies detection.

$$TCP_d = \sqrt{(TCP_x)^2 + (TCP_y)^2 + (TCP_z)^2} \tag{4.1}$$

Where, TCP_d is the Euclidean distance from the robotic arm base joint, $TCP_x, TCP_y, and TCP_z$ are the measured tool center point displacements for x, y, and z Cartesian coordinates, respectively. This DLSTM prediction model eliminates the special need of TCPs for expensive position sensors. It achieved high accuracy, as will be shown in the results section.

Figure 4.2 plots the robotic arm Tool Center Point displacement (TCP_d) Auto-Correlation Function (ACF). ACF is the correlation between TCP_d observations as a function of time lag between each observation and a prior one [46]. Response variables are corre-lated variables, as given in figure 4.2. ACF for TCP_d was calculated with the Pearson's correlation

coefficient, which is a value between [-1, 1] to describe the positive and negative correlation, and zero ACF means zero correlation [46].

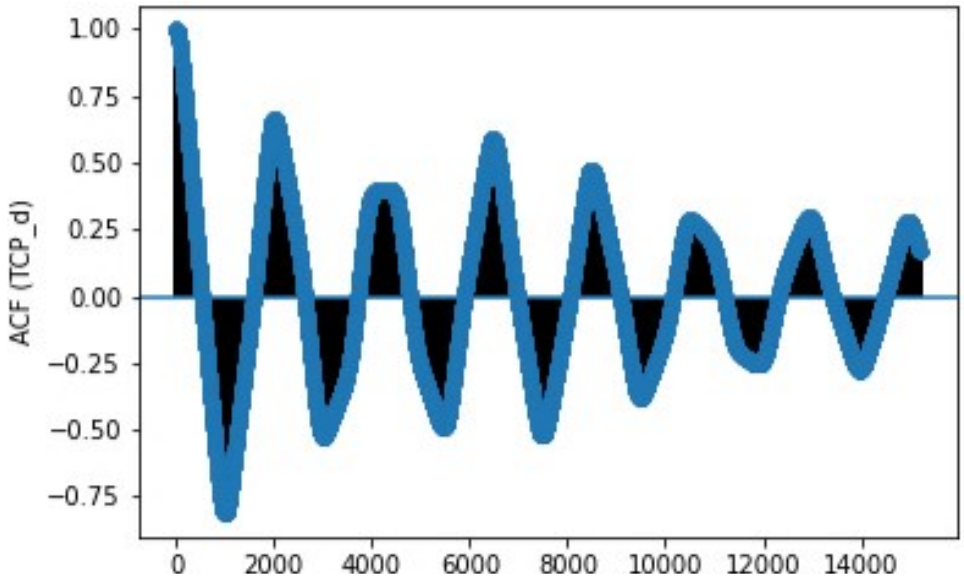


Figure 4.2 Tool center point displacement autocorrelation.

4.3.2 Multivariate Process Monitoring

Most of the industrial process has several output variables that affect process performance. Accordingly, multivariate process monitoring was developed. This was an extension of univariate quality control to avoid false alarms or failures of anomaly detection [47]. In multivariate, the monitoring data variables x is a matrix $[p \times n]$ given by $[x_1, x_2, \ldots, x_p]$, where n is number of data points and p is number of data variables. In our case, $p = 54$ variables while x is the data variables that have a variables index of $[1 : 54]$ given in Table 4.1. The monitoring variables means is $\mu = [\mu_1, \mu_2, \ldots, \mu_p]$ and the covariance matrix (Σ) is $[p \times p]$ diagonal matrix of variables' standard deviation. Therefore, the multivariate normal probability density function is given by equation 7.2.

$$f(x) = \frac{1}{(2\pi)^{p/2}[\Sigma]^{1/2}} exp(\frac{1}{2}(x-\mu)\Sigma^{-1}(x-\mu)) \tag{4.2}$$

The robotic arm TCP displacement response variable has a correlation, as shown by figure 4.2. Regression Adjustment for Multivariate (RAM) data is effective with correlated variables [48]. The RAM control chart uses linear regression to predict response variables. RAM models show the residual analysis of the predicted TCP position in comparison with the desired one.

The residual error is the difference between the predicted TCP displacement and the target value. The residuals are uncorrelated, even though the original TCP data was correlated [47, 48]. In this paper, we develop a RAM and DLSTM prediction model (DLSTM-RAM) for robotic arm monitoring. The proposed DLSTM-RAM model architecture is given by figure 4.1. RAM model has inputs of TCP target settings and DLSTM predicted values to detect the positional error anomaly with the internal parameters of residual mean and standard deviation. The DLSTM-RAM internal parameters tuning and model capabilities are discussed in results section.

4.4 results

Our DLSTM prediction model is trained with 200 training epochs, 10 input batches, 15206 training samples, and 481-validation samples. The training Mean Absolute Error (MAE) loss was 52 x 10–4. Training ended at the 30^{th} epoch when the model was trained with early stopping. Early stopping is one of the methods that we used to prevent overfitting [49]. The MAE was 27 x 10–4 with 30 epochs of training. Figure 4.3 depicts training and validation MAE losses during training with early stopping, and it shows the leaning stability of the proposed model. The lowest loss values were achieved at the 30^{th} epoch. The proposed model achieved testing MAE at 0.57% when it is tested with 8989 data samples. Figure 4.4 shows the predicted TCP displacement and the actual values with testing data. The

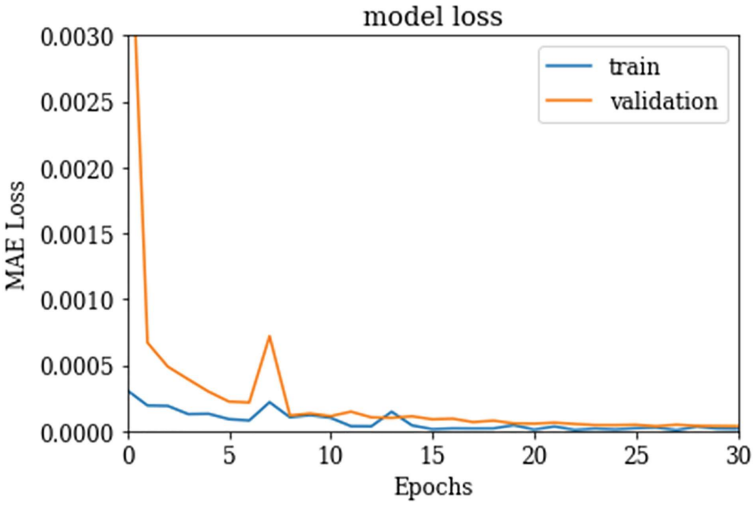


Figure 4.3 DLSTM prediction model training and validation losses with early stopping.

proposed model achieved testing MAE at 0.57% when it is tested with 8989 data samples. figure4.4 shows the predicted TCP displacement and the actual values with testing data.

For comparison, a linear regression model was designed with the same training and testing data that we used for the DLSTM model. Table 4.2 represents the MAE of our proposed model and the linear regression model for training and testing data.

Table 4.2 DLSTM model vs. linear regression prediction performance

	DLSTM	Linear Regression
MAE training (15241 samples)	27 x10-4	9 x10-3
MAE testing (8989 samples)	3.95 x10-2	13.98 x10-2

RAM detects the observation as an anomaly if the corresponding residual difference is more than the upper control limit or less than lower control limit. The control limits are defined as a residual(μ) $\pm 3^*$ the standard deviation of σ, which were calculated for the first 2000 samples data samples. Figure 4.5 shows the RAM residual control chart with testing observation data at rated speed and full payload. The number of anomaly points was 359 observations out of 6519 testing observations. The detected anomaly points are related to two positions out of the six that formulate a complete robotic arm operational cycle. The chart in figure4.5 shows a residual difference that is repeated periodically three times. Each one represents a repeated robotic arm complete operation. Therefore, the anomalies are periodically repeated. Hence, the robotic arm anomaly samples are affected by the same cause. For further research, more analysis is needed to define the causes of these anomalies. The detection response time and the proper self-healing actions requires more investigation. Self-adjustment actions can be implemented during robot operations to clear the detected anomalies.

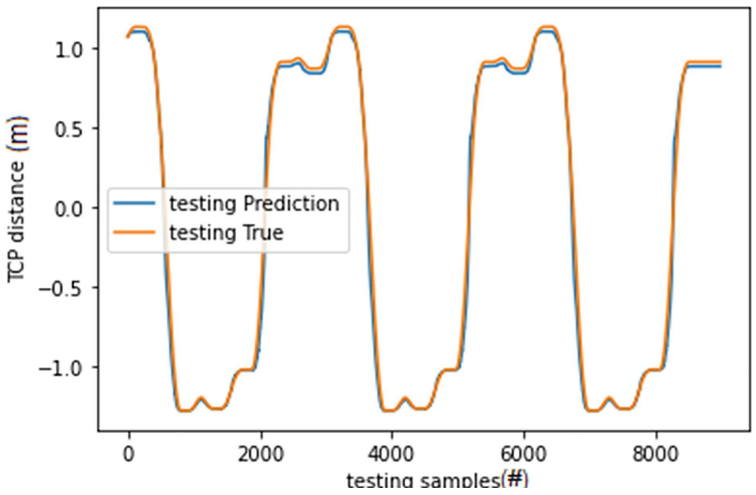


Figure 4.4 DLSTM model testing to predict the TCP distance.

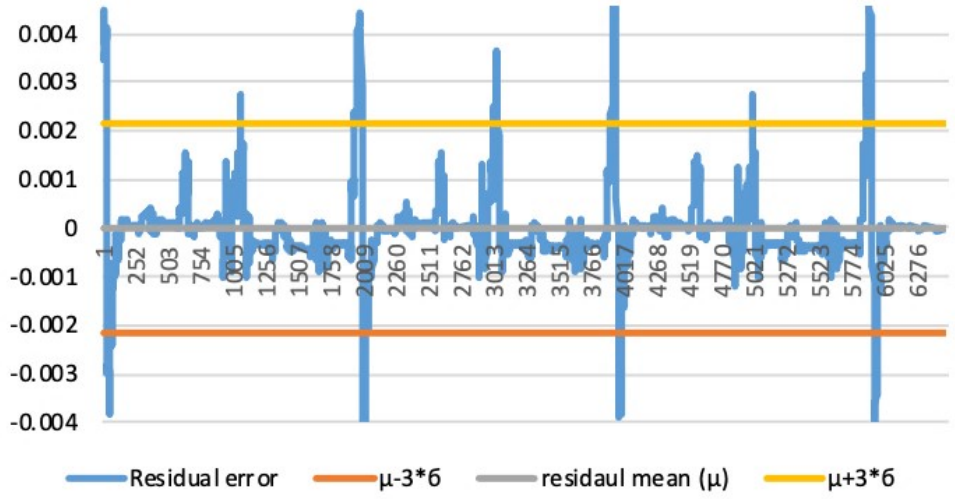


Figure 4.5 RAM Control chart for robotic arm testing data with rated speed and full payload.

4.5 Conclusion

This article presents a novel approach for robotic arm positional health assessments. Industrial robots are confronted with several environmental and field variability challenges that lead to robot failures. A combination of DLSTM-RAM models detects robotic arm TCP displacement anomalies. Our DLSTM model outperforms the traditional quality chart prediction model that uses a linear regression. It has lower MAE when testing approximately 10%. While the TCP sensor's measurement is very ex-pensive and is not available to all manufacturing plants, the proposed DLSTM-RAM detects position anomalies based on measurements taken of indicators and flags an alarm signal without the need for additional special sensors. DLSTM-RAM was tested and verified with data obtained from a robotic arm that is working at rated speed and full payload. For further research, the DLSTM-RAM model will be improved to include the position anomaly's causes, and accordingly to perform self-healing actions.

CHAPTER 5 ARTICLE 2: AUTONOMOUS SELF-HEALING MECHANISM FOR A CNC MILLING MACHINE BASED ON PATTERN RECOGNITION

Hussein A. Taha, Soumaya Yacout, Yasser Shaban
Published in: Journal of Intelligent Manufacturing.
DOI: https://doi.org/10.1007/s10845-022-01913-4

Abstract

A sustainable and reliable machining process is the main goal of seeking machine digitization. Artificial Intelligence (AI), and Cyber-Physical System (CPS) combined with Artificial Intelligence are used for process control. This has become more essential in the case of machining of high-cost aerospace materials and critical product specifications. In this paper, a novel self-healing mechanism was developed to recover a CNC machine from producing parts that do not conform to surface roughness's specifications. The machine settings are reconfigured autonomously and online to recover from the effect of tool wear and to keep the surface roughness within the design specifications. The proposed self-healing mechanism is based on a pattern recognition algorithm called Logical Analysis of Data (LAD). This algorithm generates patterns that characterize the out-of-specification state, and provides a corrective setting within the recovery patterns of the within-specification state by using various distance approaches. The developed self-healing mechanism is composed of three modules: CPS model of the CNC machine (module 1), classification into out of, or within-specification states (module 2), and a self-healing controller (module 3) that is activated if the state of out-of-specification is found by module 2. The three modules are software. The current hardware system of the machine is not altered. The proposed self-healing mechanism is validated on CNC machines with a wide range of machining parameters of feed rate from 20 mm/min to 700 mm/min and spindle speed from 10,000 RPM to 40,000 RPM. To validate the developed mechanism, a deep learning artificial model was developed on physical data to emulate the CNC milling machine in a CPS simulation environment, and test runs were executed. The proposed self-healing mechanism was evaluated under several simulation runs that covered the ranges of CNC machine settings. The measure of performance of the proposed mechanism is the out-of-specification clearing time. The validation runs show that the proposed self-healing mechanism was able to clear the out-of-specification state and to recover the within-specification state in less than three seconds, with the best distance metric approach. The results of the time response for each test run are reported.

Keywords: Autonomous machines, Process Control, Modeling, LAD, Maintenance 4.0.

5.1 Introduction

Industrial 4.0 technologies and digitization permit real-time monitoring and controlling of machine malfunctions autonomously [50, 51]. Since more information is available online and in real-time, maintenance actions can be updated adaptively to avoid process/product failure [23]. Maintenance 4.0 is a concept that exploits the connectivity between facilities, data-driven simulations, and AI techniques to develop unmanned self-healing machines [5, 18, 23]. Reliability and maintenance engineers have studied the concept of the biological immune system and call it Engineering Immune System (EIS) [5, 52]. An EIS uses AI technologies to change the classical maintenance strategy approaches that are based on the knowledge of the mean time between failures (MTBF) to the knowledge of system degradation incidences [53, 54]. This change entails continuous monitoring of sensors' readings and analytics of degradation [26, 55, 56]. To add unmanned actions to an EIS, and to achieve Maintenance 4.0, the development of software and hardware modules is needed to obtain a self-healing mechanism.

In optimization of the cutting parameters, the spindle speed and feed rate are kept constant during machining process. They are adjusted before the beginning of the machining process based on the workpiece requirements [53]. Optimization techniques have been applied to determine the optimal design settings, including speed and feed, of the CNC machine according to the condition of the cutting tool [57, 58]. The genetic algorithm was used to tune the design parameters of the CNC machine when turning AISI 4340 Steel under constraints of productivity, cost, and product specifications [57]. Practical Swarm optimization was applied to high-speed milling machines to determine cutting speed and feed rate [59]. These techniques are performed offline, and the machine is not adjusted online in response to the natural phenomenon of a process's degradation.

Fault-tolerant control (FTC) is a set of recent techniques that were developed to increase the plant availability and to reduce the risk of safety hazards. FTC aims to compensate for fault effects on the system during operation to maintain the system stability regardless of the nature of the fault. Fault-tolerant control combines several disciplines to achieve this goal by including online fault detection, automatic condition assessment, and calculation of remedial actions when a fault is detected [6, 9]. In FTC, the controller acts passively, regardless of the machine's fault nature, if it does not have a fault detection module. Huang et. al [60] designed a controller with a force estimator for a CNC milling machine to control the feed rate and to maintain the forces in a specific range while the machine is subjected to noise

uncertainties [60]. The paper addressed the online feed rate changes, but it is limited to several faults that have been defined in the design phase. The operational faults that result from the natural phenomena of degradation and the stochastic nature of the machining process are not included. In [61], adaptive control is implemented as an active FTC to control CNC milling forces online, by changing the feed rate when the measurement feedback forces deviate from an assumed value. The authors consider the force deviation as the only effect of tool wear. Sadek, et.al.2020 [62] presented a real-time tool-wear prediction to the adaptive control system as a fault detection module to improve the drilling process performance. The study is limited to one controlled variable; therefore, it has a limited recovery range, and the product specification was not addressed. Z. W. et.al (2015) developed a data-driven self-healing mechanism for a Fused Magnesium Furnace (FMF) [63]. Two software blocks were added to the original FMF system to detect the abnormalities and to control FMF while electrodes are degrading. The authors assume abnormality thresholds on the EMF's current measurement, regardless of the product specifications. To maintain the product specification, a self-adjusting CNC milling process is implemented using feed-forward Neural Network (NN) [53,64]. The model uses two interconnected NNs to predict the optimal feed rate. The first NN predicts the tool wear using cutting force sensor measurements. The second NN predicts the optimal feed rate, with the predicted wear of the first NN. The authors assume an empirical relation for the predicted feed rate and wear to calculate the surface roughness (Ra) and to choose the new machine settings. NNs are connected to sensors that measure the forces and generate feed rate directly without an anomaly detection module. In this case, the feed rate adjustment becomes very sensitive to the accuracy of the sensors' measurements. These are contaminated with noise, which is usually represented by a normal distribution (i.e., mean ± 3 SD). As such, synchronization is needed to adapt the sensor's fluctuating measurements to the real state of the milling process. In [65,66], variable online spindle speed and feed rate were applied to high-speed milling machines to stabilize the product quality. These solutions require adding special sensors to track the variation of Ra, and system stability analysis is required to verify the new machining parameters. The neural network self-healing provides an uninterpretable machine setting that is highly sensitive to the NN's hyperparameters tuning. Shaban, et.al (2017) provided an online alarm system that is triggered upon the detection of a malfunction or a quality failure according to the patterns generated by LAD. The results demonstrate that LAD outperforms the NN in detecting machine anomalies [20]. They linked the uncontrollable sensors' measurements to failure by using pattern recognition. During online processing, the fault is detected based on the current sensor's measurements and the extracted patterns of malfunction. They did not provide a self-healing mechanism in case of fault incident.

By definition the self-healing mechanism detects the system's function-failures, and autonomously executes recovery actions without disrupting its operation [5,15,63,67]. The function failure could be caused by either system anomalies and/or component failure [11,15,18]. In case of physical damage, the self-healing systems need to have a system redundancy through the system's connected redundant components to overcome the physical damage without interrupting the system's operation. [6,11]. With the capability of detecting system anomalies or faults, the self-healing actions do not change the physical structure or the interconnections of the system component [18,20]. It only allows the system to recover and to continue its operation without disruption.
This paper introduces a new approach for an autonomous self-healing mechanism of CNC milling operation based on the pattern recognition. The proposed self-healing mechanism deals with product nonconformity faults and machining process anomalies. The main contribution of this paper is the design of a self-healing mechanism that has the following properties:

1. It does not change the hardware of the machine.
2. It uses a pattern recognition algorithm to generate patterns that differentiate between the within and out-of-specification states. This algorithm is not based on any empirical formulation, but machine learning principles.
3. It is triggered to avoid producing products that do not conform to specifications, and it is applied online.
4. It has a fault detection module to provide warnings once the product moves out-of-specification
5. It is complemented by a synchronization module that is added to adjust the response time.

Section 2 of the paper presents the mechanism of the self-healing system. Section 3 describes the methodology used to build the self-healing mechanism, including the pattern generation algorithm (LAD), the detection of an out-of-specification state, and distance approaches. Section 4 demonstrates the implementation process that was built with a deep learning artificial model to emulate the CNC milling machine. Section 5 presents the validation process and the obtained results. Section 6 presents conclusion and future work.

5.2 The Self-healing Mechanism

In the context of autonomic systems, the self-healing module/component detects the system malfunctions and performs corrective actions without disrupting the system. [5,15,63,67].The proposed self-healing mechanism is developed to recover a machine from a state that would lead to the production of out-of-specification products by changing the values of the controlled variables of the machine. This is a dynamic mechanism that adapts to the actual machine's status online. The mechanism is composed of the closed-loop that is provided in figure 5.1. This loop enables the self-healing mechanism to interact with the CNC machine to prevent it from producing out-of-specification products. The mechanism includes three modules. Module 1 is the CNC milling machine that will ingest the LAD algorithm with the sensors' measurements and the controllable variables' settings of the machine. Module 2 is the fault-detection module that will analyze the data, apply the LAD algorithm to generate patterns that characterize the within and the out-of-specification status, and classify the recent sensor's readings' levels as belonging to one of these two states. The patterns are generated by LAD and online modelled by IF-THEN-Rules. Each decision rule represents one of the extracted patterns, which are generated by LAD. Module 2 provides online monitoring and analysis of the product quality at each time step (t). In the case study of this paper, Module 2 diagnosis the system status, and raises a fault flag for an out specification status. The fault detection module works with the CNC machine's model in synchronous mode, and it reads the machine sensors' values at each time step (t); therefore, the fault detection response is dependent on the sensors' measurement response time. Module 3 is the self-healing module. If the out-of-specification fault is detected at time step (t), the Self-healing module reads the current machining parameters, then finds the nearest recovery pattern from the generated patterns based on a distance approach (r). To close the autonomic loop, the self-healing actions are set on the CNC machine in Module 1. At the next time step (t+1), fault detection Module 2 states whether the fault is uncleared. If so, a new loop of data ingestion and analysis, pattern generation, and self-healing execution are taken. A synchronization block is added to Module 3 to avoid the impractical oscillations of the new settings and to stabilize the autonomic loop.

5.3 Materials and Methods

This section presents the LAD algorithm to generate the patterns, which are used to develop the self-healing actions in Module 3, and to classify and detect the out-of-specification in

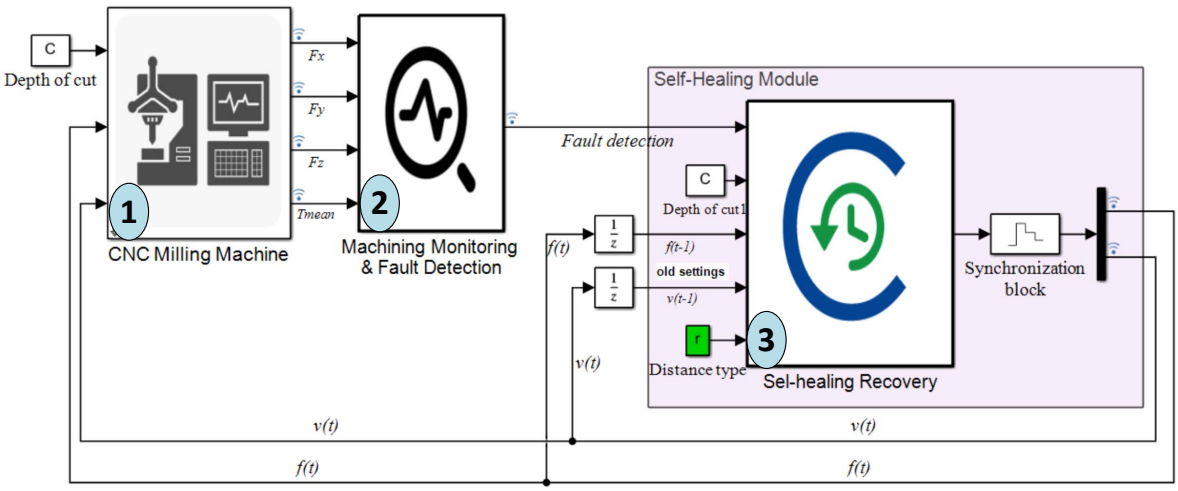


Figure 5.1 CNC machine Self-healing closed-loop system

Module 2.

5.3.1 Logical Analysis of Data and pattern generation

Logical Analysis of Data (LAD) is a supervised data mining technique that combines Boolean logic functions and combinatorial optimization to solve the classification problems [32]. LAD generates explanatory logical patterns, which are defined by data features' boundaries to identify each state of the machine. Yacout et al. developed a cbmLAD software for condition-based maintenance applications with LAD [32]. LAD generates patterns that segment the state space into zones that belong to different classes. Pattern generation is the step of finding the logical relation between the data's features to characterize these zones. Each logical relation represents a pattern, and each pattern covers a range of data observations within a specific range of a feature values. cbmLAD utilizes a Mixed-integer Linear Programming (MILP) algorithm to solve pattern generation problems that characterize positive (faulty) Ω^+ and negative (non-faulty) Ω^- classes, in the case of two classes' classification problem. The MILP is given by the optimization problem 1, where a pattern's index y_i is zero if the observation i is covered by pattern p, and 1 otherwise. In a matrix form, i is the data observation number and j is the feature that is transformed into attributes of binary form b_j and b_{j+n} $\forall j \in [n, 2n]$ are the complementary values of b_j. The decision variables are w_j, a binary index that takes the value 1 if the attribute j exists in the generated pattern, and zero otherwise, d is the pattern degree that specifies the number of attributes in the generated

pattern, and n is the total number of binary attributes.

$$
\begin{aligned}
\min_{i\in\Omega^+} \quad & \sum y_i \\
\text{s.t.} \quad & w_j + w_{n+j} \leq 1, \forall j = 1, 2, ..n \\
& \sum_{j=1}^{2n} a_{i,j} w_i + n y_i \geq d, \forall i \in \Omega^+ \\
& \sum_{j=1}^{2n} a_{i,j} w_i \leq d - 1, \forall i \in \Omega^- \\
& \sum_{j=1}^{2n} w_i = d \\
& 1 \leq d \leq n \\
& w_j, y_i \; \textit{are binary}, \; \forall i, j
\end{aligned}
\tag{5.1}
$$

The generated pattern is defined by the attribute b_j when $w_j = 1$ and by its complement when $w_{j+n} = 1$. cbmLAD solves the MILP model in 5.1 until all of the data observations in the positive class are covered by at least one pattern, and then reruns it for the negative class observations. In this paper, positive patterns are the patterns that indicate an out-of-specification state, and negative patterns represent the within-specification zones.

5.3.2 Distance Metrics Approach

To find the pattern that changes the machine status from an out-of-specification status to a within-specification status, two distance approaches are implemented to find the nearest within-specifications pattern to the current pattern that is detected in Module 2 of figure 5.1. These two approaches are point-to-point distance and point-to-distribution. Consequently, the Self-healing module will automatically generate the new machine settings that are specified in the nearest within-specifications' pattern observed at time (t+1) to the previous existing settings at step (t).

Point-to-point (P2P) distance approach

The P2P distance equation is provided in Equation 7.2 [68,69], where the ${D^r}_p$ is the distance between the machine settings vector x at time step (t) and m is the mean vector of each recovery pattern p. The point-to-point distance type r is the Manhattan distance if $r = 1$,

and the Euclidean distance when $r = 2$. k is the index of the controllable variables.

$$D^r{}_p = (\sum_{k=1}^{2} |x_k - m_k|^r)^{1/r} \tag{5.2}$$

Point to distribution (P2D) approach

To implement the P2D, the Mahalanobis Distance is used. It is calculated from the CNC settings at the time step (t), at which point the out-of-specification status is detected. The Mahalanobis distance $D^{r=3}{}_p$ is calculated with Equation 3 [70,71], where x is the vector of the machine settings at time step (t), m is the vector of mean values of each controllable variable in the recovery pattern and Cov is the covariance $[k \times k]$ matrix of the controllable variables in a recovery pattern. In this study, the P2D distance approach considers all possible settings inside each recovery pattern, as they have the same priority while calculating the covariance matrix Cov. The number of recovery settings to be considered in the calculation of the Cov matrix influences the self-healing algorithm execution time, as shown in the appendix.

$$D^{r=3}{}_p = \sqrt{(x-m)^T cov^{-1}(x-m)} \tag{5.3}$$

As an example of the implementation of this methodology, Table 5.1 shows a sample of machining data for a product's delamination under different machining settings of spindle speed [1500-12000] in RPM and feed rate [20-800] in micron/rev. The exit delamination is within-specification when its value is lower than, or equal to, 1 and out-of-specification otherwise. To build a self-healing module for this problem, the given data in Table 5.1 is ingested to cbmLAD to generate the positive (faulty) and negative (corrective non faulty) patterns. In this example, cbmLAD generates four corrective patterns, as given by Table 5.2.

Table 5.1 Example of CNC exit delamination data

v/f	20	60	100	200	400	600	800
1500	1.00	1.00	1.00	1.08	1.11	1.22	1.24
5000	1.00	1.00	1.05	1.05	1.06	1.14	1.17
8500	1.05	1.00	1.00	1.05	1.06	1.10	1.11
12000	1.00	1.00	1.00	1.04	1.06	1.04	1.07

When the delamination's out-of-specification state is detected, the self-healing mechanism selects the nearest recovery pattern according to the distance approach r. For example, if the fault is detected when the machine settings were 8500 RPM and 600 micron/rev, the self-healing mechanism calculates the distance between these faulty settings and each

Table 5.2 cbmLAD Generated Negative Patterns for the delamination quality characteristic

N.o	Spindle speed	Feed rate	Delamination
1	$v < 5000$	$f < 200$	Within-specification
2	$v < 8500$	$f < 100$	Within-specification
3	$8500 < v < 12000$	$f < 200$	Within-specification
4	$v > 5000$	$20 < f < 200$	Within-specification

of the four recovery patterns in Table 5.2. The nearest recovery pattern is the pattern that has the shortest distance to the current faulty settings according to the distance approach r. In case of P2P, the mean of each pattern is equal to $[m_1, m_2, m_3, m_4]$ where $m_1 = [v_{1|min} + \frac{v_{1|max} - v_{1|min}}{2}, f_{1|min} + \frac{f_{1|max} - f_{1|min}}{2}]$. The mean values are m_1= [3250,110], m_2= [5000,60], m_3= [10250,110], and m_4= [8500,110]. When r=1 the Manhattan distances are equals to D_1^1=75.76, D_2^1= 63.56, D_3^1= 47.33, and D_4^1= 22.13. The shortest distance is D_4^1 and it indicates the 4^{th} recovery pattern, which is shown in Table 5.2, as the nearest recovery pattern to the current faulty settings of [8500, 600], hence the self-healing selects the 4^{th} pattern to generate the corrective settings with a P2P distance approach.

5.3.3 Machining Monitoring and Fault Detection

Module 2 activates/deactivates the self-healing mechanism where it monitors the CNC machining process. At each time step (t), Module2 reads CNC machine sensors data and analyzes it to decide that the current machining status conforms to the required specifications. Once an undesired machining performance is detected, Module 2 activates the self-healing mechanism in module 3 of figure 5.1. In the case of multi-quality factors, Module 2 diagnoses the fault type in addition to the instant of the detection.

In this paper, LAD is used to monitor the machining process and to detect the non-conformed products. LAD outperforms other Machine Learning algorithms where it generates explanatory patterns that defines the fault and its causes [20, 21]. Each pattern is a logical relation among the machine's sensors (uncontrollable variables), the current measurement is said to be covered by a pattern when it satisfies the pattern logical relation. In online mode, each generated pattern is represented by an IF-Then-Rule and this rule defines the current machining status.

To explain the online implementation of Module 2, Table 5.3 shows the sensor data of the example given in Table 5.1. It contains the product exit delamination and the corresponding uncontrollable variable measurements of the forces x-direction $F_x(N)$ and mean-temperature $T_{mean}(C^o)$. For pattern generation, the data example in Table 5.3 are ingested to cbmLAD

to extract the explanatory patterns that are capable to monitor and define the machining process status. It is stated that the working piece conforms when its exit-delamination is lower than or equal to one. Table 5.4 summarizes the generated patterns for the data example

Table 5.3 Delamination data example for uncontrollable variables

F_x/T_{mean}	100	150	200	250	300	350	400
10	1.00	1.00	1.00	1.08	1.11	1.22	1.24
20	1.00	1.00	1.05	1.05	1.06	1.14	1.17
30	1.05	1.00	1.00	1.05	1.06	1.10	1.11
40	1.00	1.00	1.00	1.04	1.06	1.04	1.07
50	1.07	1.00	1.00	1.03	1.05	1.05	1.05

in Table 5.3. Four patterns represent the conforming delamination product and another four patterns for the out-of-specification product. Each pattern is a logical relation between the forces F_x and mean-temperature T_{mean}, and it is bounded by a range of sensors' measurement values. In online mode at any time step (t), one of these logical relations is satisfied and its related pattern is defined, then the current machining status is concluded. The innovation aspect of this approach is the interpretable patterns that indicate the cause of receiving this status. For example, if the sensor reading at a time (t) is [50 N, 230 C^o] for F_x and T_{mean} respectively. The product delamination is out-of-specification as this measurement is covered by the 1^{st} pattern of the non-conforming patterns. The high temperature is the main cause to have this non-conformed working piece. To build module 2 that triggers the self-healing mechanism for delamination out-of-specifications, the four non-conforming patterns in Table 5.4 are modeled by four If-Then-Rules.

Table 5.4 Extracted cbmLAD's Patterns to characterize the product delamination based uncontrollable variables

N.o	$F_x(N)$	$T_{mean}(C^o)$	Delamination
1		$T_{mean} > 255$	Out-of-specification
2	$F_x > 45$	$T_{mean} < 120$	Out-of-specification
3	$15 < F_x < 25$	$T_{mean} > 175$	Out-of-specification
4	$25 < F_x < 35$	$T_{mean} < 125$	Out-of-specification

5.4 The implementation of the self-healing mechanism to a milling process

The proposed self-healing system in figure 5.1 is applicable to any machining process with different materials and different quality factors. The developed mechanism is validated to recovers the CNC machining faults with a wide range of controllable variables for feed rate [20 mm/min - 700 mm/min], and spindle speed [10,000 RPM - 40,000 RPM]. This mechanism can be applied to all engineering systems, which have controllable variables that can be manipulated automatically.

In this paper, the self-healing mechanism is tested on a CNC machine while routing a carbon fiber reinforced polymer material. The experimental data and the data collection procedures are described in [20]. An example of the collected data is given in Table 5.5. The CNC milling machine's experimental raw data is defined by four controllable variables: feed rate f, speed v , tool length , and depth of cut C, and sensors 'measurements of forces, F_x, F_y, F_z, and mean temperature T_{mean}. At the time of data collection, the surface roughness (Ra) is physically measured on the product after machining time and it was categorized as within-specification (0) or out-of-specification (1) according to the Ra value [20]. To emulate the developed self-healing mechanism, we build and validate an artificial CNC milling machine model. Table 5.5 shows a sample of the experimental data. Each observation is classified based on the value of produced surface roughness (Ra) value, whether it is within-specification or out-of-specification. The collected data is imbalanced because it consists of 100 observations with an out-of-specification Ra and only 8 observations that have within-specification values. To balance the data, the AMSCO algorithm in [72], is applied to generate new observations within the specification values of the Ra.

A deep neural network model is built to simulate the CNC milling machining process in CPS. It has three inputs: the depth of cut C(mm), feed rate f (mm/min), and spindle speed v (RPM), and four outputs, 3D forces (N) and the mean temperature (C^o). To increase the learning accuracy, we design more than 10,000 different NN architectures with different layers that varied from one-layer models to five-layer models. The number of hidden neurons varied from 2 up to 9 neurons. The best model architecture to be selected is the model that achieves the lowest Mean Square Error (MSE) for unseen testing data.

The sensors' measurement vector $[F_x, F_y, F_z, T_{mean}]$ of the physical machine is the output of the NN model, and we apply 10-folds cross-validation to find out the best NN model architecture. The NN modeling experiment's details and the architecture that has the minimum MSE for each combination of neurons and layers are given in figure 5.2. With the lowest MSE of 311, the best NN architecture consists of four hidden layers with hidden neurons of 9, 6, 4, and 6 neurons for the $1^{st}, 2^{nd}, 3^{rd}, and\ 4^{th}$ layers, respectively.

Table 5.5 Sample of The Experimental Raw Data (Shaban et al., 2017)

No	v $X10^4$ RPM	f Mm/min	C (mm)	Tool length	F_x(N)	F_y(N)	F_z(N)	T_{mean}(C^o)	Ra fault
1	4	250	32	38	9.2	5.8	6.5	305.031	1
2	4	500	32	38	15.4	11.2	6.6	385.058	1
3	4	1000	32	38	25.5	20.5	11.5	437.552	1
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
13	4	250	32	31	9.3	5.8	6.4	231.178	1
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
25	4	1000	32	24	29.5	17.6	11.1	421.388	1
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
32	4	250	64	38	11.6	7.1	9.1	292.493	1
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
66	4	250	96	38	14.2	7.8	3.1	417.309	1
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
100	1	1000	96	24	79.5	96	49.1	405.448	1
101	4	250	32	32	18.4	5.7	3.2	305.491	0
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
107	2	250	64	24	24.2	11.2	6.1	220.27	0
108	3	250	96	24	23.2	8	4.9	281.012	0

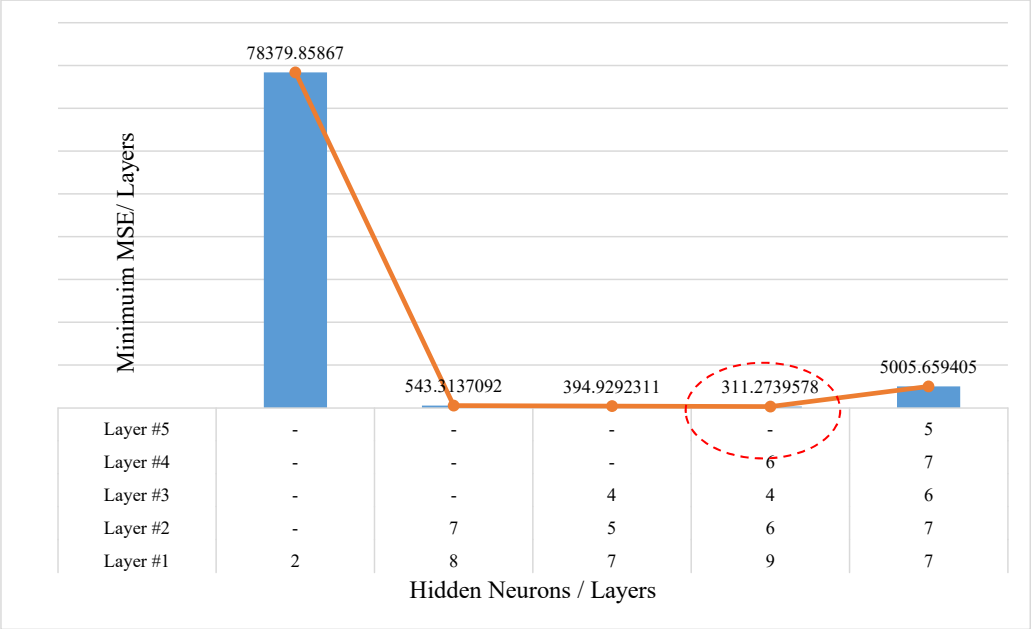


Figure 5.2 Minimum MSE per each model's layers for unseen testing data

To build the NN model, we used the best NN architecture to be trained and tested on the deep TensorFlow environment [73]. With the developed deep learning model, The Mean Absolute Error (MAE) is 5.477 for training and 6.747 for testing (e.g. physical measurement = model reading $\pm$ 6.747 C^o). Figure 5.3 shows the uncontrollable variables' values of $(a)F_x, (b)F_y, (c)F_z$, and (d) T_{mean} obtained with the NN model versus the actual physical measurements for testing data. Once the CNC machine has an out-of-specification Ra, the proposed Self-healing algorithm produces a corrective action based on the recovery patterns that are given in Table 5.6 to clear this fault. As a result, the faulty state is updated in the next time step. To evaluate this interaction in the Cyber-Physical System (CPS), we developed an artificial CNC milling machine model and validated this model with the physical raw data in Table 5.5.

5.4.1 Generation of The Recovery Patterns

The recovery pattern matrix contains patterns that are obtained with the software cbmLAD [32] when the CNC machine product is within Ra specifications. These patterns define the zones of controllable variables; the spindle speed, the feed rate, and depth of cut values that would recover the machine from an out-of-specification status. These patterns are extracted by using the Logic Analysis of Data (LAD) algorithm. Table 5.6 shows the six

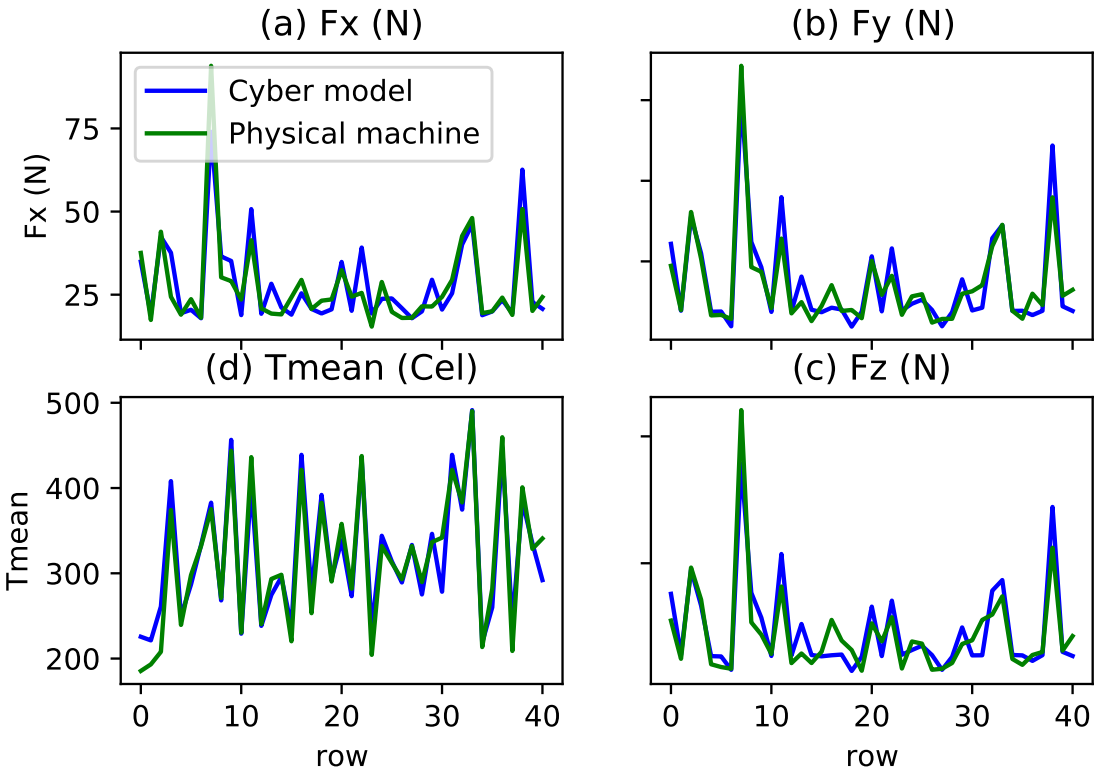


Figure 5.3 uncontrollable variables of $(a)F_x, (b)F_y, (c)F_z, and\ (d)T_{mean}$ with the NN model vs. actual physical testing data

recovery patterns in terms of the controllable variables that are generated by the cbmLAD software [32] from the data shown in Table 5.5, columns 2 to 5. These patterns classify the milling process according to the Ra values that are within-specifications. Two MILP

Table 5.6 Recovery Patterns for Machine controllable variables Generated by cbmLAD.

Pattern	v ($\times 10^4$RPM)	f(mm/min)	C(mm)
1	$v > 1.5$	$f < 375$	$C < 48$
2	$v > 3.5$	$f < 375$	$C < 80$
3	$v > 3.5$	$f < 750$	$C < 48$
4	$2.5 < v < 3.5$	$f < 375$	$C > 80$
5	$1.5 < v < 2.5$	$f < 375$	$C < 80$
6	$1.5 < v < 2.5$	$f < 750$	$C < 48$

problems are solved by cbmLAD. First, the controllable machine's settings that are shown in Table 5.5, columns 2 to 5, are used to extract recovery (negative) patterns that are shown in Table 5.6. The second problem is to detect the Ra's fault within one of the positive patterns

that are generated in terms of the uncontrollable variables. The values of these variables are shown in columns 6 to 9 of the sensors' measurements in Table 5.5.

5.4.2 Out of specification detection modeling

To eliminate additional hardware, the product quality monitoring and fault detection is developed with using of the current machine sensor data. The fault detection module is modeled the extracted patterns that generated by applying the Logical analysis of data (LAD) algorithm to the CNC milling machine sensors' data, which are shown in columns 6 to 9 in Table 5.5. These patterns segment the multidimensional space that is formed by the forces and the temperature's values into two zones: conformed or non-conformed Ra to specifications. An observation of CNC machine sensors' data is classified according to the patterns generated by the uncontrollable variables of forces and temperature. To classify and detect the nonconformity of the working piece online at each time step (t), each one of the generated positive patterns is represented by an If-Then-Rule. The machine measurements are forces (F_x, F_y, F_z) in Newton (N) and mean temperature in Celsius (C^o). The fault detection rules generated by the cbmLAD tool are as follows:

1. IF ($F_x > 24.7$ N) THEN non-conformed Ra
2. IF ($T_{mean} > 366.474$ C^o) THEN non-conformed Ra
3. IF ($F_y > 5.75$ N) AND (225.724 $C^o < T_{mean} < 361.878$ C^o) THEN non-conformed Ra
4. IF ($F_x > 22.4$ N) AND ($F_y > 5.75$ N) AND ($F_z > 3.05$ N) AND (190.028 $C^o < T_{mean} <$ 361.878 C^o) THEN non-conformed Ra
5. IF ($F_x > 18.15$ N) AND ($T_{mean} > 225.724$ C^o) THEN non-conformed Ra
6. IF ($F_x > 14.9$ N) THEN non-conformed Ra
7. IF ($F_y >$5.75 N) AND ($T_{mean} < 203.86$ C^o) THEN non-conformed Ra

5.4.3 The Self-healing module

At each time step (t), the fault detection module reads the machine sensor's measurements and generates a fault flag signal in the case of non-conforming Ra. The fault flag signal is transferred to the Self-healing module. In this case, the self-healing algorithm that is shown in Table 5.7 uses the P2P or the P2D approaches to find the closest recovery pattern, out of those that are shown in Table 5.6 in terms of feed and speed. This algorithm recovers the

machining process from the out-of-specification state by returning it to within-specification state, while minimizing the setting fluctuations to avoid process instability. According to pattern selection, the self-healing actions at time (t+1) is a vector S= $[v_{t+1}, f_{t+1}]$, whose value is between the minimum values $[v_{min}, f_{min}]_p$ of the selected pattern p, and the maximum values of the vector $[v_{max}, f_{max}]_p$.

In this implementation example, the developed Self-healing algorithm with the P2P distance approach shows the same performance when using the Manhattan distance ($r = 1$) and the Euclidean distance ($r = 2$) for all experiments. The difference between the results becomes significant with a large number of variables [69]. In this study, we have only two controllable variables, spindle speed and the feed rate, since the depth of cut (C) value is set according to the design of the product before the machining process begins. Thus, the self-healing mechanism deals with it as a constant value while recovering the Ra from the out-of-specification state. The self-healing actions are the recovery settings of feed rate (mm/min) and spindle speed (RPM) at time step (t).

Table 5.7 Propose self-healing algorithm of the CNC Milling Process

Algorithm: Self-Healing Mechanism of CNC Milling Machine

1. Load cbmLAD pattern matrix and record recovery class (-ve) patterns $p \in \Omega-$
2. Reorder recovery patterns matrix in ascending order according to C of each $p \in \Omega-$
3. **While** (machine Running) do:
4. **While** (fault detected) do:
5. **Case** P2P distance
6. Calculate the distance between $[v_t, f_t]$ and $[v^{mean}, f^{mean}]_p$ for distance type $r \in [1,2]$.
7. **Case** P2D distance ($r=3$)
8. Calculate the Mahalanobis distance *between* $[v_t, f_t]$ *and* pattern's *Cov*
9. **Select** working patterns according to C.
10. **Select** pattern p^* with the smallest distance D_p and according to distance case.
11. Set $[v_{t+1}, f_{t+1}]$ where $[v^{min}, f^{min}]_{p^*} < [v_{t+1}, f_{t+1}] < [v^{max}, f^{max}]_{p^*}$
12. **END** for each
13. **END** for each

The algorithm in Table 5.7 recovers the surface roughness (Ra) fault and provides online automatic actions to ensure that it meets the product specifications. The input vector for the Self-healing module contains the machine settings at a time (t) and the fault detection signal, in terms of forces and temperature. Meanwhile, the output vector contains the recovery settings of feed rate (f_{t+1}) in mm/min and spindle speed (v_{t+1}) in RPM. To save time

during the online computation, the self-healing algorithm will be active only if there is a detected fault. It is structured on the Ra recovery patterns in terms of controllable variables, fault detection in terms of the force and temperature, and the distance metric approaches, which will be discussed in the next section.

5.5 Validation of the Self-healing Mechanism

To validate the performance of the self-healing system, machine settings that were not used in the training by the NN algorithm were chosen. These settings are listed in Table 5.8. They represent the maximum and the minimum values that are allowed for the controllable variables, which are the depth of cut, the feed rate, and the speed, according to the recommendation of tool's supplier. The combination of these values leads to 27 observations of settings. These observations are ingested to module 1 in figure 5.1 to find the estimated class for each observation according to the seven patterns of If-Then-Rules that were shown previously in the modeling of module 2.
Next, the performance of the self-healing algorithm is evaluated by using the SIMULINK simulation environment, and the sampling frequency is 20 kHz. Seven settings out of the 27 initial observations lead to conforming Ra as shown in Table 5.8. These sets are excluded from the next steps, since the objective is to simulate the self-healing mechanism by beginning with a setting that will lead to out-of-specification states. Before simulating the autonomic closed-loop in figure 5.1, the CNC machine is set with the initial settings listed in Table 5.8 for the 20 out-of-specification labels. In 4 settings, the P2D distance approach led to better recovery time. These are settings 2,7, 8, and 9 from Table 5.8. In 9 settings, both approaches led to comparable recovery times. These were settings 18 to 27 from Table 5.8. Finally, 7 settings let to better recovery time with the P2P approach. These are settings 3, 4, 6, 10, 11, 12, 15 from Table 5.8. In the following section, we present the synchronization mechanism using setting 2, 7, 8 and 9. The results of the other settings are shown in the Appendix.

5.5.1 The Self-healing mechanism Synchronization

In the 2^{nd} initial set, and with a P2P distance approach, the self-healing module receives the Ra fault flag; then it starts to interact with the CNC machine model to adapt to the nearest recovery settings of feed rate and spindle speed based on the recovery patterns in Table 5.6. Figure 5.4.b and c present the actions of the P2P self-healing module to clear the detected CNC milling Ra flag. Time (seconds) is on the x-axis, while the y-axis presents the machine controllable variables of the depth of cutting C(mm), feed rate f(mm/min), and spindle speed $v(\times 10^4$ RPM), respectively, and the detected fault flag is in figure 5.4.d.

Table 5.8 The 27 Observations of the CNC machine's Initial Settings with Fault detection (module 2) response

Run N.o	C mm	f mm/min	v $\times 10^3$ RPM	Ra Specifications state
1	32	100	10	Within
2	32	400	10	Out
3	32	700	10	Out
4	32	100	20	Out
5	32	400	20	Within
6	32	700	20	Out
7	32	100	40	Out
8	32	400	40	Out
9	32	700	40	Out
10	64	100	10	Out
11	64	400	10	Out
12	64	700	10	Out
13	64	100	20	Within
14	64	400	20	Within
15	64	700	20	Out
16	64	100	40	Within
17	64	400	40	Within
18	64	700	40	Out
19	96	100	10	Out
20	96	400	10	Out
21	96	700	10	Out
22	96	100	20	Out
23	96	400	20	Out
24	96	700	20	Out
25	96	100	40	Within
26	96	400	40	Out
27	96	700	40	Out

The red rectangles in figure 5.4.b and figure 5.4.c are 5 $m$second time windows. They depict the high fluctuating self-healing actions of feed rate and speed. The fault flag was not cleared during the fluctuating period, at the same time the self-healing module generates new recovery values every t= 5 μseconds. This fluctuation is not acceptable in a practical implementation, because it causes instability in the machining operation, with the possibility of releasing the Ra flag once again. To deal with this drawback, the proposed self-healing module and the fault detection module is synchronized. In The case study of this paper, the machining material is carbon fiber composites, and the thermocouple lead wire is the temperature sensor as demonstrated in [74, 75]. The temperature sensor's measurement increases with

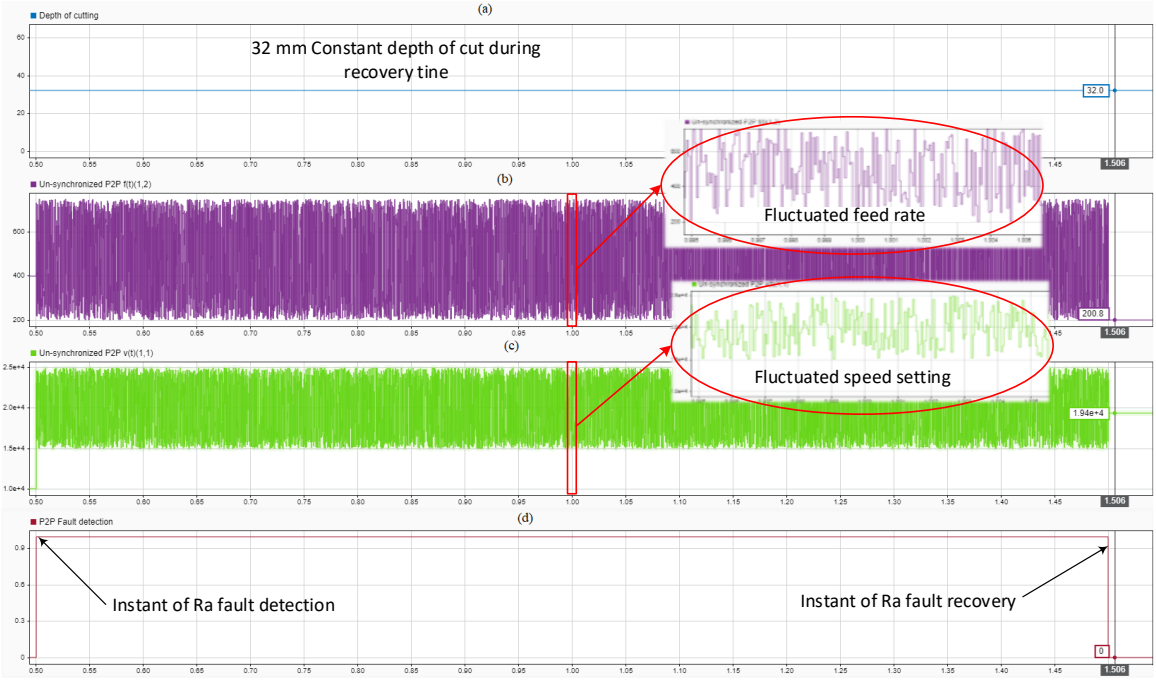


Figure 5.4 Unsynchronized P2P-Self-healing algorithm actions while recovering from Ra faulty state;(a) depth of cut, (b) feed rate, (c) Speed, and (d) fault detection flag.

time, and this increment is more rapid as the feed rate increases. At 250 mm/min feed rate, thermocouple lead wire reads the value of the maximum temperature before 0.5 second [74]; consequently, we model the measurement sensors with a response delay of 0.5 second. We use this 0.5 second delay to synchronize the Self-healing module 3 in figure 5.1, to the fault detection module 2 and CNC machine module 1.

The Self-healing module actions of recovery feed rate and speed are synchronized to the fault detection module by holding the values of the actions until the next update of the sensor measurements and fault detection signal, and we adjust the synchronization block to 0.5 second. The P2P synchronized self-healing actions are presented in figure 5.5. The fault detection signal is given in figure 5.5.a and is cleared after 1.5 seconds with the self-healing action of adjusting the feed rate from 400 mm/min to 200.8 mm/min in figure 5.5.c, and the spindle speed from 1×10^4 RPM to 1.94×10^4 RPM in figure 5.5.e. P2P distance approach selects the 6^{th} recovery pattern in Table 5.6 to clear the fault because it is the nearest pattern to the feed's actual setting. Consequently, the sensor reading of F_x goes from a value of 36.9 N to 16.6 N, as determined by figure 5.5.d. In figure 5.5.f, the mean temperature decreased from 241 C^o to 225.9 C^o.

With the 2^{nd} initial settings and the use of the P2D distance, the synchronized Self-healing mechanism demonstrates faster recovery than the P2P approach. The P2D Self-healing

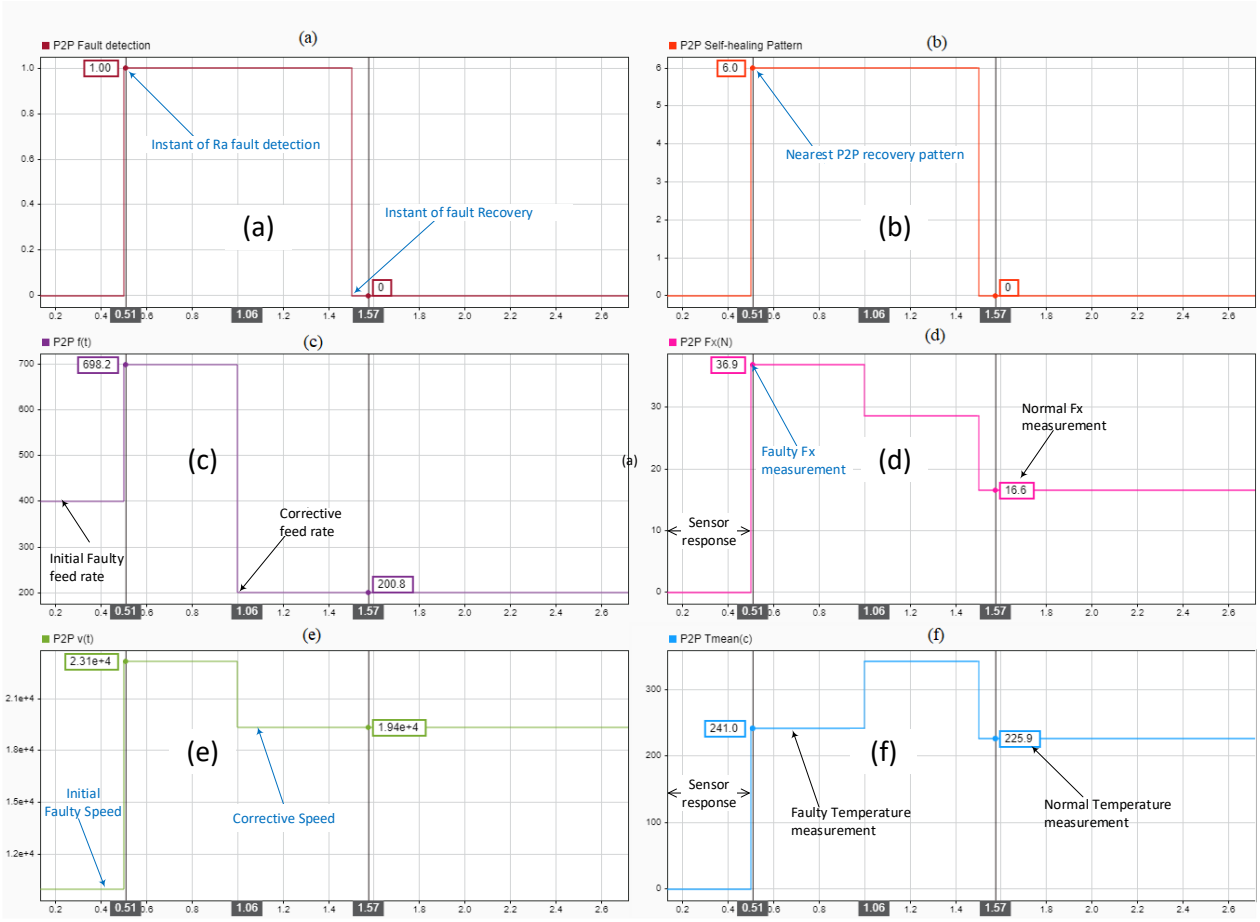


Figure 5.5 P2P- synchronized Self-healing module interacting with CNC milling fault with the 2^{nd} initial setting.

algorithm selects the 1^{st} recovery pattern in Table 5.6 as the nearest recovery pattern, as shown by figure 5.6.b. The self-healing module takes 1 second to recover the CNC machine and to clear the fault detection signal in figure 5.6.a. The F_x is the sensor's reading that activated the fault detection module, and its value decreased from 36.9 N to 19.4 N when the recovery pattern was implemented, as shown in figure 5.6.d. Machining settings were changed from 400 mm/min to 329.5 mm/min for the feed rate, and from $1{\times}10^4$ RPM to $2.08{\times}10^4$ RPM for the spindle speed.

Table 5.9 summarizes the evaluation of the synchronized self-healing mechanism for the initial settings 2, 7, 8, and 9 before and after the recovery. We note that, for these 4 settings, the P2D approach performs better than the P2P in terms of recovery time. Among the test-runs in Table 5.8, the longest recovery time is 18 seconds and it is executed by the P2P self-healing module with 7^{th},8^{th}, and 9^{th} initial settings. The detailed results of these runs are in Appendix A. It should be stated that with this longest recovery time, the proposed

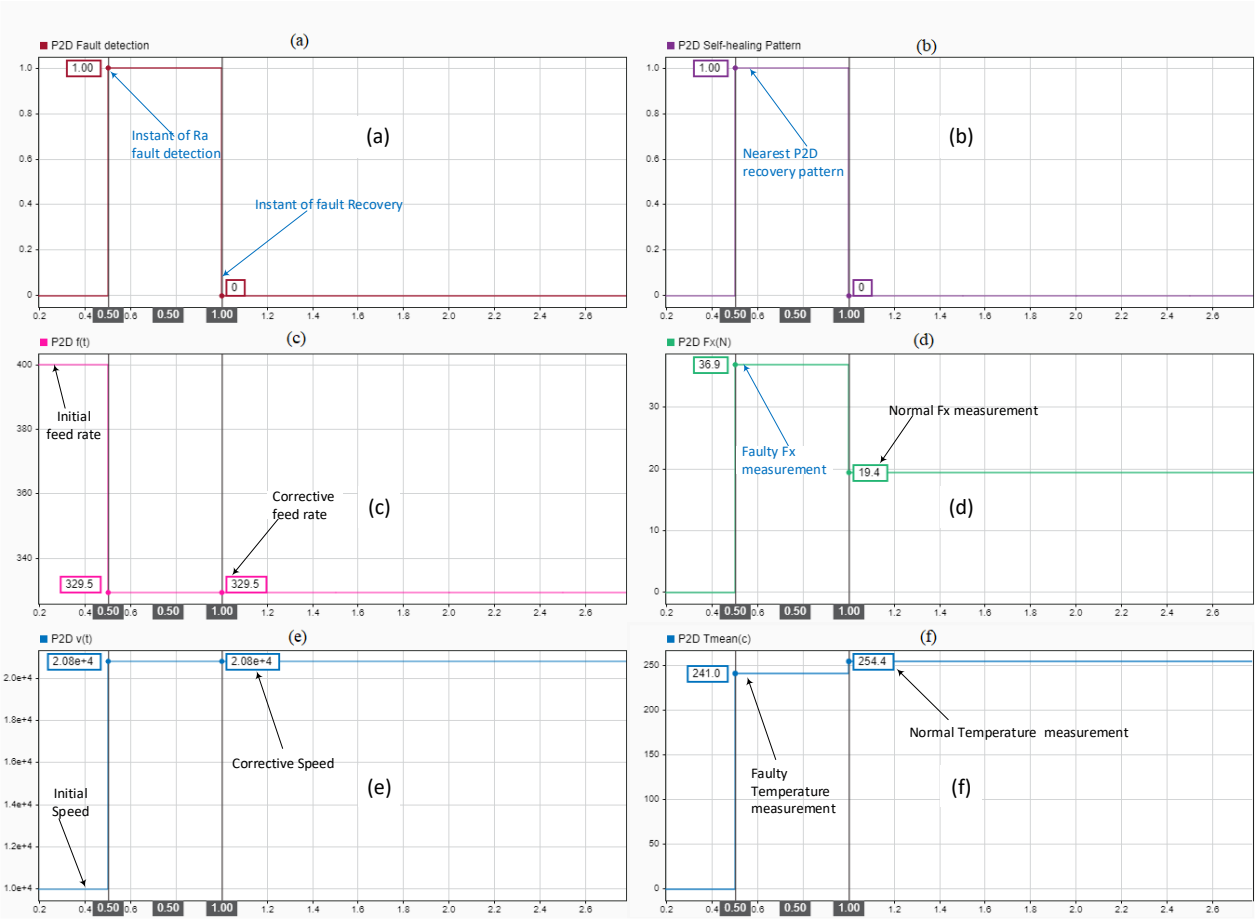


Figure 5.6 P2D-synchronized Self-healing module interacting with CNC milling fault in setting 2

Table 5.9 Summary of the Synchronized Self-Healing Mechanism for 2, 7, 8, and 9 Initial Settings

N.o	Distance approach	Initial settings		Sensors' readings Before recovery		Final Recovery settings		Sensors' readings After recovery		Recovery time (s)
		f mm/min	v $\times 10^3$ RPM	Fx(N)	$Tmean(C^o)$	f mm/min	v $\times 10^3$ RPM	Fx(N)	$Tmean(C^o)$	
2	P2P	400	10	36.9	241	200.8	19.4 $\times 10^3$	16.6	225.9	1.5
	P2D					329.5	20.8 $\times 10^3$	19.4	254.4	1
7	P2P	100	40	16.9	285.5	499.3	35.6 $\times 10^3$	18.3	280.8	18
	P2D					308.2	31.3 $\times 10^3$	15	243.5	3.5
8	P2P	400	40	17	284.6	499.3	35.6 $\times 10^3$	18.3	283.5	18
	P2D					308.2	31.3 $\times 10^3$	15	243.5	3.5
9	P2P	700	40	22.5	331.7	499.3	35.6 $\times 10^3$	18.3	283.5	18
	P2D					308.2	31.3 $\times 10^3$	15	243.5	3.5

self-healing mechanism still outperforms the classical approaches and improve the machining process. In the classical techniques, the product surface roughness is measured offline after

machining the part and if it is found to be non-conformed, this part has to be reworked. In this case, the CNC machine is stopped to change its parameter before milling the next products [20]

5.6 Conclusion

This article proposes a novel self-healing mechanism based on pattern recognition for an autonomous CNC milling machine. The objective is to produce products that are within the predefined specifications. In this paper, we apply the proposed mechanism to a milling process of carbon fiber reinforced polymer material, where the product quality is defined by surface roughness (Ra). The fault detection module is built on the pattern recognition of faulty observations, which are obtained from sensors' readings of 3D forces and temperature at each time step (t) and raises a flag when these readings exhibit patterns of out-of-specification zones. Once a faulty state is detected, the self-healing mechanism changes the controllable machine variables of feed rate and spindle speed to return to the within-specifications zones of surface roughness. The proposed mechanism is achieved by building a Self-healing module that interacts with the CNC machine and fault detection module in online operation. The self-healing mechanism is based on P2P and P2D approaches that search for the nearest patterns that are generated by solving a MILP of the Logical Analysis of Data (LAD), which defines the within-specification zones for the controllable variables of feed and speed. The implemented Self-healing algorithm uses the explanatory recovery patterns that are generated by cbmLAD software [32] and the two distance approaches. To recover the CNC machine's faulty state, the corrective actions are set by manipulating the controllable variables, the feed rate, and the spindle speed, at time step (t+1) based on the nearest recovery pattern to the nonconformity machine settings at time (t). While the Self-healing module interacts with the CNC machine, the forces and mean temperature measurements of the CNC machine are reduced to the conformity ranges. Consequently, the fault detection module is cleared.
Synchronizing the Self-healing module with the fault detection module prevents the fluctuation of Self-healing actions that lead to an unstable autonomic loop and instability of the machinery process. Consequently, it would lead to the inability to clear a fault.
The proposed synchronized Self-healing module is evaluated with 27 initial machine settings that cover the possible ranges of the CNC machine's controllable variables. These settings are initialized the CNC machine's cyber model during its testing phase. Twenty runs out of twenty-seven lead to out-of-specification Ra initial settings. The self-healing algorithm with a Point-to-Distribution (P2D) distance approach shows three kinds of performance over twenty simulation runs. In **the first performance**, the P2D-self-healing has faster fault

recovery than the P2P-self-healing with four settings. Moreover, the P2D distance approach selects different recovery patterns from the P2P approach. The maximum P2D-Self-healing recovery time was 3.5 seconds. In **the second performance**, the P2D Self-healing module has the same performance as P2P in nine runs out of 20-runs. In **the third performance**, Self-healing selects the same recovery pattern either with P2D or P2P distance approaches, yet, the P2D recovery time of the CNC machine out-of-specification Ra is slower than P2P. The computation time of P2D distance depends on the number of corrective data points used to calculate the pattern covariance matrix. This matrix is more accurate with more corrective points, but it increases the computation time on the other side. Self-healing was evaluated for all 27 runs with 1000 corrective points to calculate the P2D distance. When the number of corrective samples decreases to 10 samples, the P2D Self-healing execution time becomes faster and it performs the same as P2P Self-healing.

The Self-healing algorithm selects the corrective machine settings from a uniform distribution of the spindle speed and feed rate values with the same priority over the recovery pattern range. In future work, we will address the setting values that could achieve smooth machine transitions, low energy consumption, and high productivity. In addition, the variations of spindle speed affect the machining stability [65], a stability analysis is acquired to be added to the self-healing algorithm. Moreover, the algorithm evaluation will be extended to cover different working materials and machining types.

CHAPTER 6 ARTICLE 3: AIRCRAFT ENGINE REMAINING USEFUL LIFE PREDICTION FRAMEWORK FOR INDUSTRY 4.0

Best Graduate Paper Award, IEOM, Toronto 2019

Hussein A. Taha, Ahmed. H. Sakr, Soumaya Yacout
Published in: The 4^{th} North America IEOM proceedings

Abstract
This article proposes a Condition-Based Maintenance (CBM) approach for aircraft engines and Remaining Useful Life (RUL) monitoring, and failure prevention. Due to the unavailability of run-to-failure data, Turbofan Engine Simulation data, obtained from NASA repository, is used to train and test our model. Data Acquisition and Management system framework and planning are proposed for online monitoring and RUL prediction. In practice, sensor measurements usually suffer from noise contamination, hence the prediction models are challenged by noise contaminated data for both training and testing tasks. This is done to assess their prediction ability in a similar condition of having noisy data. Linear and nonlinear prediction models are developed, with performance comparison addressing both regression and classification problems. Models performance indices consider both prediction accuracy and percentage of predictions before the actual failure (PBAF). The proposed model considers continuous learning and improvement to account for any further operational changes that affect the model prediction ability. This is reached by ingesting the model with the actual RUL during the maintenance of the engine unit, and by comparing it to the predicted one.
Keywords: Condition-based maintenance, Failure prediction, Engine Degradation, IoT, Industry 4.0..

6.1 Introduction

Aircraft engine is a critical component. Its failure causes loss of lives. The traditional maintenance strategies, that are proposed by the designers, usually involve Reliability Centered Maintenance (RCM). These strategies propose preventive maintenance tasks that are based on reliability analysis of the operating systems. These strategies improve effectively the reliability of the engine. However, the costs are high due to unnecessary maintenance or replacement actions. Condition-Based Maintenance (CBM) is used for cost minimization while achieving reliability improvement. Online monitoring and data analysis lead to better maintenance planning and maintenance duration reduction. In addition to performing effective

maintenance plans, airlines can achieve better consistency of flight scheduling.

CBM is a condition monitoring concept which is used to decide when the operating asset requires maintenance [7, 76]. This provides a proactive scheduling for the maintenance process. The CBM strategy begins with data acquisition from sensors' readings, which are analyzed to extract useful information about the system's state [7]. The performance of CBM is challenged by data cleanness and prediction models' accuracy [7, 77]. Normally, an engine condition should trigger maintenance actions within enough time before failure. Consequently, efficient models that accurately predict the RUL are required while overcoming the noise contamination problems [78]. Researchers proposed supervised learning prediction models for aircraft engine degradation [27, 77, 79–83]. However, their models do not consider continuous learning, hence there is no possibility for accuracy improvement or considering any new events that the model was not trained for. In practice, industrial operations usually have operational modifications that require continuous monitoring to avoid inaccurate predictions [84]. The online monitoring of operating assets has become possible through Internet of Things (IoT) technologies adopted by the Industry 4.0 paradigm. These give a chance for sensors to transmit the captured engine data to a cloud database during operation [84]. The cloud storage of the data facilitates the engine monitoring even if the aircraft is in the air. Hence, maintenance scheduling is achieved, and flight rescheduling is planned to avoid conflicts. Our proposed framework consists of:

- Data acquisition and cloud storage platform
- RUL prediction model

In this paper, the data that is used for training and testing of the prediction models, is obtained from NASA Prognostics Data - Turbofan Engine Degradation Simulation results [78]. Simulation is used due to the difficulty of having run-to-failure real data for these engines. This article is organized as follows: System planning, and framework layout are presented in section 2. Data prepossessing and overview of the prediction models are given in section3. Section 4 discusses the obtained results. Finally, section 5 presents our conclusion and future works.

6.2 System planning and framework layout

The data represents simulation results for 100 engine units. It is provided by a text file of 26 columns and indexed into units, cycle time, three types of operational settings, and 21 sensors' measurements. Each row is a snapshot of the data that is taken during a single

operational cycle. Table 6.1 shows detailed description of sensors' measurements. The actual RUL for an operational cycle is the difference between the unit's total life until failure, and the current cycle's number. It is calculated for the training and the testing data as the failure cycle for each unit is given by its simulation results

The main objective for a condition-based maintenance strategy is to predict the number of remaining operational cycles before failure, i.e the number of operational cycles after the current cycle, during which the engine will continue to operate. However, this prediction task is challenged by data contamination due to sensor noise. The measurements types are summarized as follows:

- Temperature measurement
- Pressure measurement
- RPM measurement
- Air Mass flow measurement

A data acquisition system is needed for transfer and storage of the sensors' measurements. Aircrafts have data acquisition system with aviation Arinc429 standard [85]. It is used to transfer data such as air data, radar altimeter data, and GPS data. The measurements are used for engine operational control [86]. Our proposed system layout includes sensors' measurements data transfer to an onboard server as shown in Figure 6.1. The server is selected with internet/cloud connecting feature; thus, it facilitates the engine remote monitoring and RUL prediction, even when the aircraft is in operation.

6.3 Methodology

The methodology that is applied for model training and testing is performed using Scikit-learn library for machine learning on Python 3.7. Python is an open-source general-purpose programming language. The Scikit-learn is a free machine learning library that features various classification and regression algorithms. The Python code loads the input data from CSV file. The CSV file is developed from the raw text file using MS Excel.

6.3.1 Data prepossessing

The preprocessing of the data is an important step before training machine learning models. Some problems within the data, such as correlated predictors, presence of outliers, missing

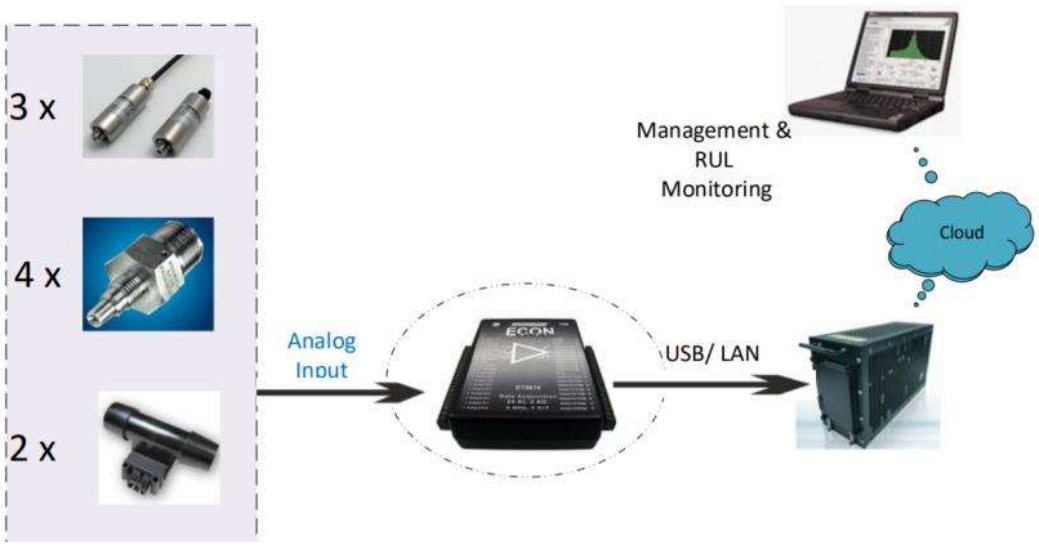


Figure 6.1 Proposed system layout

Table 6.1 Descriptions of sensor signals

Index	Predictor name	Unite
1	Total temp fan inlet	K^o
2	Total temp LPC outlet	K^o
3	Total temp HPc outlet	K^o
4	Total temp LPT outlet	K^o
5	Pressure fan inlet	psia
6	Total pressure in bypass	psia
7	Total pressure HPC outlet	psia
8	physical Fan speed	RPM
9	physical Core speed	RPM
10	Engine Pressure Ratio	-
11	HPC outlet preasure	psia
12	Fuel flow Ratio to "11"	PPS/psi
13	corrected fan speed	RPM
14	corrected Core speed	RPM
15	bypass ratio	-
16	Fuel air ration	-
17	Bleed enthalpy	-
18	Demand fan speed	RPM
19	Demand Core speed	RPM
20	HPT coolant bleed	ibm/s
21	LPT coolant bleed	ibm/s

data instances, cannot be handled well by some machine learning techniques and may affect their prediction capabilities. Hence, it is advisable to preprocess the data to improve the performance of the models. The preprocessing applied here includes the following:

- Outliers detection and removal
- Removing highly correlated predictors

The outliers are detected by Box plot. The data instances that have a Z-score higher than 3 are considered outliers and are removed. Figure 6.2 depicts sensor 7 data as an example for outlier removal. The data instance that is red colored has a Z-score greater than 3. This instance is removed from the input data. The same procedure is applied for the other predictors.

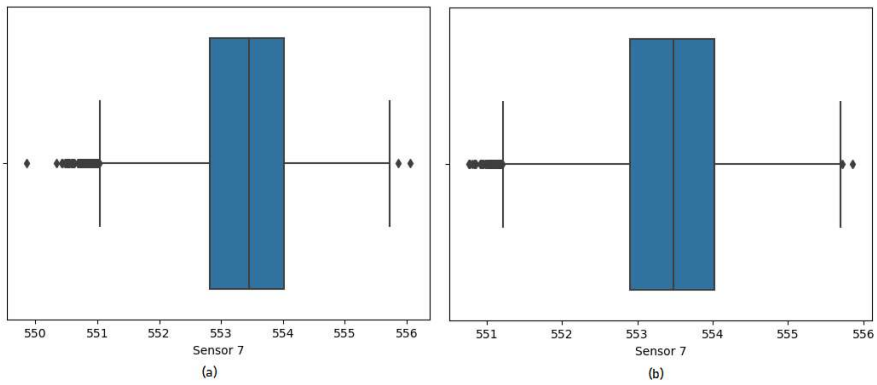


Figure 6.2 Box Plot for sensor 7 before (a) and after (b) removing the outliers

Figure 6.3.a shows the correlation matrix for the predictors. The matrix represents the coefficient of correlation between each of the predictors and the others. This coefficient ranges from -1 to 1. The sign defines the type of proportionality between the predictors. The relationship is directly proportional for a positive coefficient of correlation, while is inversely proportional for a negative one. Large absolute value of the coefficient of correlation, greater than 0.95, shows high correlation. The values are color coded to aid visualization. Highly correlated predictors, Setting 3, Sensor 1, Sensor 5, Sensor 10, Sensor 16, Sensor 18, and Sensor 19, are removed from the input data before models' training as shown in Figure 6.3.b.

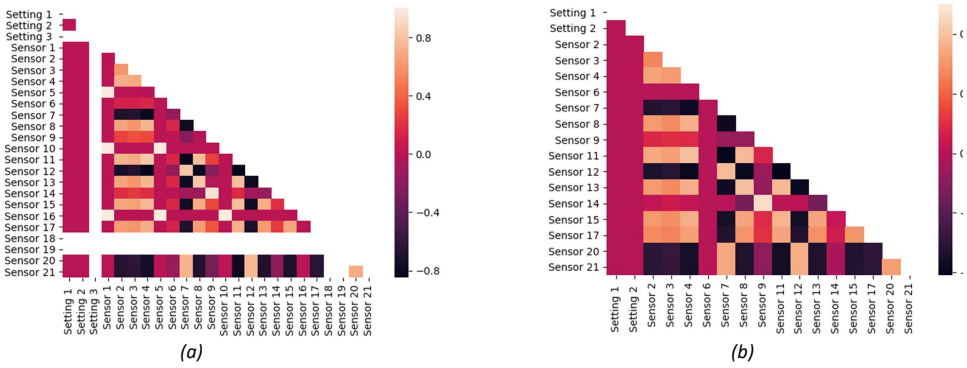


Figure 6.3 Correlation matrix for (a) all predictors, (b)removing highly correlated predictors

6.3.2 Prediction Models

To predict the RUL, both linear and non-linear models are explored including parametric and non-parametric types. Different transformations for the output are tested in order to select the best form for RUL prediction. The best form is selected based on the prediction performance of the models. The performance is measured by the root mean square error (RMSE) for predictions using the testing data. For this data, the best form for the output is the inverse form, 1/RUL, for all the tested models. The input data is standardized to eliminate the effect of the predictors data units on the prediction models. The explored models include the following:

- Linear parametric:
  - Multiple linear regression
  - Ridge regression
  - Partial least square regression (PLS)
- Non-linear parametric:
  - Polynomial regression
- Non-linear Non-parametric:
  - K-nearest neighbors (KNN)
  - Random Forest (RF)
  - Neural Networks (NN)

Equation 6.1 presents the multiple linear model where y_p is the predicted RUL value according to the transformation that is applied for the RUL of training data,X_j is the j^{th} predictor, P is the number of predictors which is 17 for the input data after removing highly correlated ones, and β_0, β_j are model parameters. The Ridge regression model is shown by Equation 6.2 where λ is the Ridge parameter. A value of 0.2 is selected for this parameter based on the best performance for prediction. The polynomial model, degree 2, is given by Equation 6.3. This degree is selected to avoid the overfitting problem that the polynomial model suffers from when the degree is high. The overfitting results in low training error, but high test error and poor prediction ability. This problem is named as the bias-variance trade-off in literature [87]. The Ridge and the PLS models are explored for their ability to control and reduce the regression coefficients variance, hence improving the prediction performance. The Ridge model involves shrinking the coefficients towards zero, while the PLS considers dimensions reduction for the predictors [87].

The KNN regression model is given by Equation 6.4 where K is number of neighbors, and F_i is inverse of the distance between two neighbors. y_i is the RUL value, according to the applied transformation, for i^{th} nearest data point to the given X. The number of neighbors K is selected to be 5 according to the best prediction accuracy found. This avoids the overfitting problem as small K values are avoided. For the Random Forest model, the best parameters for prediction performance using the testing data are 100 trees with a depth of 20. The square root of predictor number is considered when looking for the best split. The Neural Network model consists of three hidden layers with sizes of 10, 8, 4, and the activation is rectified linear unit (ReLU). The size of hidden layers is selected to be between the size of the input layer and the output layer as recommended in [88]. Model performance is assessed by RMSE which is shown by Equation 6.5 [87].

$$y_p = \beta_0 + \sum_{j=1}^{p} (\beta_j X_j) \tag{6.1}$$

$$y_p = \beta_0 + \sum_{j=1}^{p} (\beta_j X_j) + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{6.2}$$

$$y_p = \beta_0 + \sum_{j=1}^{p} (\beta_j X_j) + \sum_{j=1}^{p} (\beta_j X_j^2) \tag{6.3}$$

$$y_p = \frac{F_i}{\sum_{i=1}^{K} (F_i)} \sum_{i=1}^{K} (y_i) \tag{6.4}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i}(y_i - y_{pi})^2} \tag{6.5}$$

6.4 Results

The prediction of the RUL may have an error which results in a prediction of failure before the actual failure, PBAF, or after the actual failure, PAAF, as shown in Figure 6.4. Both cases are considered error from the point of view of RUL prediction. However, having a predicted life which is beyond the actual life is worse than having a prediction which is shorter than the actual life. For this, the performance of the models is measured not only based on the value of error, but also based on the PBAF%. The best-case scenario, in this case, is having the least possible value of error, along with the highest PBAF%.

6.4.1 Regression Method

The regression is done in this context to predict the value of the RUL. Figure 6.5 shows the performance measurement of the selected models based on RMSE and PBAF%. As shown in Figure 6.5, the Random Forest regressor is the most suitable over studied models with the lowest RMSE and high PBAF%. The Neural Network regressor gives the highest RMSE value among all models. The Neural Network yields the highest PBAF%, nearly 70%, while the Random Forest yields fewer PBAF%, up to 58%.

Figure 6.6 shows the relative importance of predictors based on the Random Forest model as the best model in this case. In Random Forest, the decrease of the Residual Sum of Squares (RSS) at each split is recorded. The predictor that has the highest value of RSS total reduction in all splits is the most important. The predictors importance gives better understanding for the most important engine readings that are affected by the RUL of the

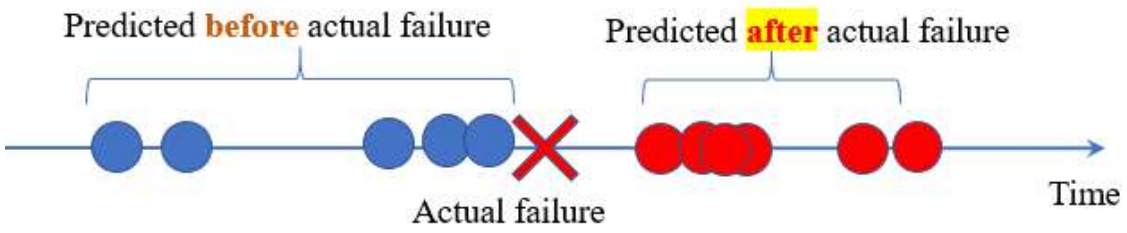


Figure 6.4 Positions of possible error in RUL within time axis

engine. The figure shows that Static pressure at HPC outlet, Sensor 11, is the most important predictor for RUL prediction. The Total pressure in bypass-duct, Sensor 6, has no importance as shown in the figure. The physical sensor measurement is no longer required; hence the amount of measurement data size are reduced accordingly.

6.4.2 Classification Method

Due to the less satisfactory results of the explored regression models, an alternative methodology is proposed which involves classification of 2 RUL classes instead of directly predicting its exact value. The RUL values are transformed into percentages for each engine unit, then a class is assigned for each data instance based on the RUL% value. This value is assigned according to the desired maintenance strategy. For demonstration, the RUL% value that differentiates the classes is arbitrarily selected to be 20%. The classes are as follows:

- Class 1: RUL is more than 20 %
- Class 2: RUL is less than 20 %

Three different classifiers are tested at different classification thresholds:

- logistic regression
- KNN classifier (5 neighbors)
- Random Forest classifier (Depth=8, 50 trees, Max features → "Sqrt")

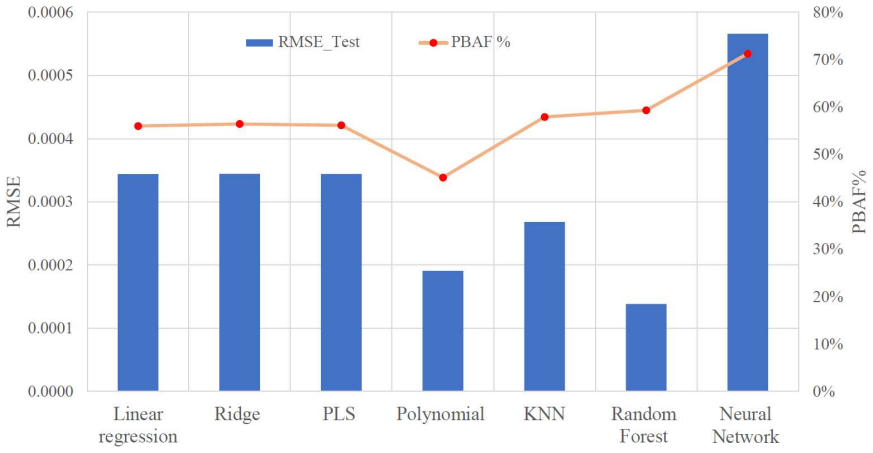


Figure 6.5 RMSE and PBAF% for the proposed prediction models

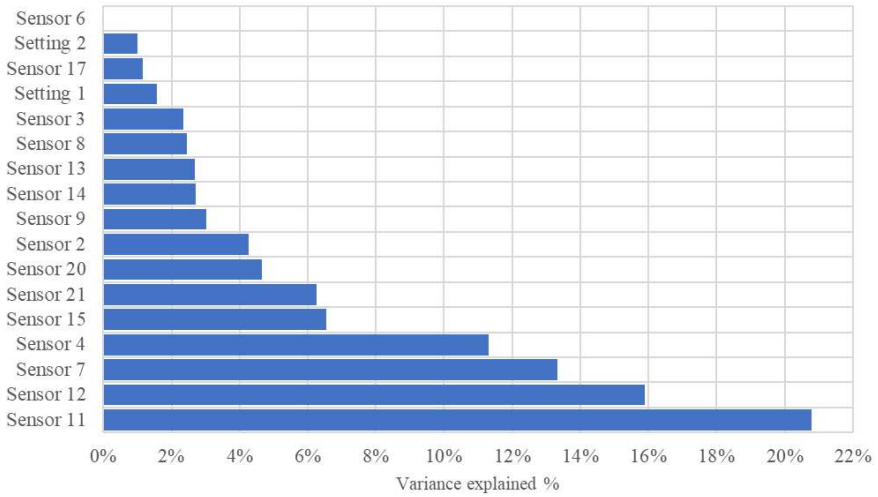


Figure 6.6 Predictors relative importance based on Random Forest model

The number of neighbors in the KNN and both the depth and number of trees in the Random Forest are selected according to the best class prediction performance found.
Figure 6.7.a shows both the error rate and the PAAF%. The error rate represents the proportion of the false classifications obtained for the test data. The Random Forest classifier gives the minimum error rate and PAAF%. Classification methods calculate the probability of selection for each class and perform the selection according to its classification threshold. This threshold affects the classification error rate and the percentage of false classification in each class. The false classification in a certain class is changed when the threshold is modified [87]. The PAAF% is not acceptable error type, and it is reduced by decreasing the classification threshold as shown in Figure **??**.b. Although the error rate has increased for all classifiers, the Random Forest shows a promising result as the PAAF% is successfully reduced to only 1.24% at 7.43% general error rate.

Figure 6.8 shows the execution time in seconds for different parts of the Python code that is used for the application of our methodology. These times are based on 2.5 GHz Core-i5 CPU with 8 Gb of RAM. The data loading and preprocessing is shown in green bar. The regression models are shown in blue bars. The classification models are shown in orange bars. The Random Forest classifier takes less execution time for training as compared to the Random Forest regressor. This is due to that the classifier has less depth and number of trees than the regressor.

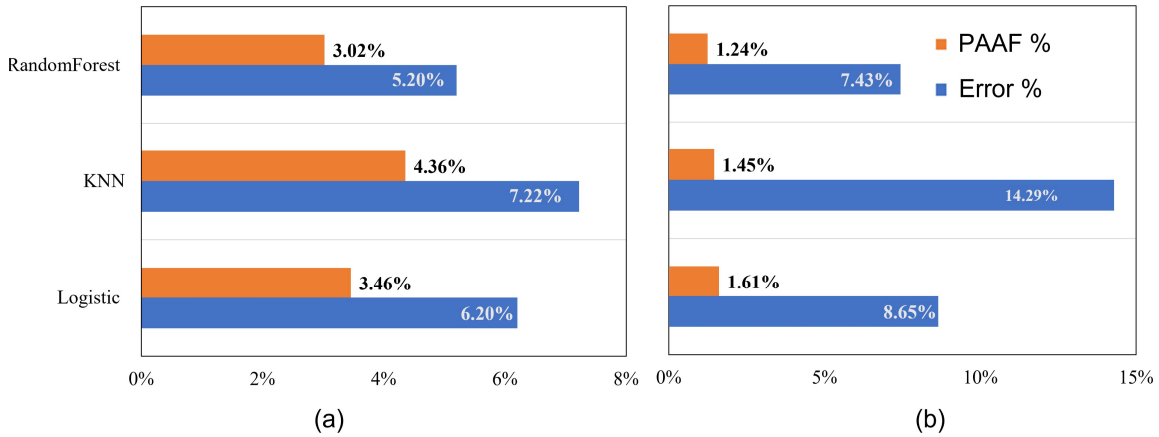


Figure 6.7 Error rate and PAAF% at classification threshold of: (a) 0.5, and (b) 0.2

Figure 6.8 Execution times for different models using Python

6.5 Conclusion

This research proposed a framework for aircraft engine's RUL prediction. This framework included On-line remote monitoring and continuous learning with cloud connection facility. The RUL prediction model parameters are meant to be updated every maintenance operation, which helps improving the accuracy and the predicting capabilities of the model. The sensor noise problem was overcome by our model which affirms its robustness. This promotes

its ability to provide reliable predictions with real data that is normally contaminated with noise. The input data were preprocessed before exploring the prediction possibility. The preprocessing included outliers and highly correlated variables removal for reaching better modelling performance. We studied both regression and classification methodologies for performing RUL prediction. The Random Forest classifier showed promising results. It offers safe and conservative condition-based maintenance. It could provide RUL classes prediction, above/below a certain level. This was demonstrated at 20% level of RUL. The classes prediction was achieved at only 1.24% PAAF% and 7.43% general error rate.

CHAPTER 7 ARTICLE 4: DEEP REINFORCEMENT LEARNING FOR AUTONOMOUS PRE-FAILURE TOOL-LIFE IMPROVEMENT

Hussein A. Taha, Soumaya Yacout, Yasser Shaban

Under Review in: International Journal of Advanced Manufacturing Technology.

Abstract

This paper develops an approach to improve a CNC machine's tool performance and slow down its degradation rate automatically in the Pre-Failure stage. A Deep Reinforcement Learning (DRL) agent is developed to optimize the machining process performance online during the Pre-Failure interval of the tool's life. The Pre-Failure agent that is presented in the proposed approach tunes the feed rate according to the optimal policy that is learned in order to slow down the tool's degradation rate, while maintaining an acceptable Material Removal Rate (MRR) level. The machine learning techniques and pattern recognitions are implemented to monitor and detect the tool's potential failure level. The proposed mechanism is applied to a CNC machine when turning Titanium Metal Matrix Composites (TiMMC). A CNC machine Digital Twin (DT) is developed to emulate the physical machine in the digital environment. It is validated with the physical machine's measurements. The proposed pre-failure mechanism is a model-free approach, which can be implemented in any machining process with fewer online computational efforts. It also covers a wide range of cutting speeds, up to 15,000 RPM. Deployment of the proposed machine learning approach for the particular case study improves the tool's Time to Failure (T2F) by 40% and the MRR by 6%, on average, compared to the classical approach.

Keywords: Degradation rate, Potential Failure, P-F Curve, Reinforcement Learning, Tool performance.

7.1 Introduction

The integration of Artificial Intelligence (AI) in a Cyber-Physical System (CPS) is used to establish autonomous and self-driven machining processes [23, 24, 89]. The machining processes are usually associated with aspects of non-linear behavior and stochastic degradation that result in the difficulty of predicting the life span of the tool, especially when dealing with difficult-to-cut materials [21, 90]. There are many attempts in the literature to monitor the tool wear and detect the machining's tool failure. To achieve the highest possible material

removal rates in machining, previous studies focused on offline optimization to schedule the machine feed rate, assuming ideal machining conditions [57]. Despite the previous attempts, an intelligent online system that can monitor and optimize the tool's performance in real-time is still needed. This paper fills this gap by offering a new approach that provides an intelligent-based extension of Tool Time to Failures (T2F) while maintaining an acceptable material removal rate level.

Offline mathematical optimizations were applied to the CNC machine processes to find the static machining parameters that maximize productivity [91]. Feed rate scheduling optimizations were developed to have a dynamic online feed rate setting [64, 90, 92]. One of the limitations of these approaches is the usage of empirical equations that assume ideal machining conditions. Adaptive control (AC) techniques take into consideration the environmental and sensor variations by mathematically estimating the forces at each time step, and then comparing the estimated values with the actual sensor measurements [90]. As such, the CNC machine controller's parameters are changed to achieve the offline optimized feed-rate schedule. The estimation of online forces requires large computational time. Stemmler, et.al [93] developed a Model Predictive Controller (MPC) to minimize the production time online for CNC milling machines. MPC is a model-based controller that predicts the values of the forces and adjusts the feed rate online accordingly to minimize the machining time. The MPC online optimization causes a processing delay, and thus, an additional signal processing synchronization is added to the machine controller. Both MPC and AC require mathematical modeling to estimate the forces. These models assumed a new tool at the beginning of cutting and ideal tooling conditions.

Shaban et.al studied tool wear monitoring for CNC machines to develop a failure alarm solution that could avoid producing defective pieces [20, 56, 94]. The authors applied Logical Analysis of Data (LAD) to detect the tool wear (VB) failure while monitoring the data of the machining forces. Sadek et.al 2020 [62] developed an adaptive mechanism that linked the tool wear monitoring to AC for a drilling machine. This mechanism is limited to two speeds and two feed rate adjustment levels. Shaban et.al used the time to Failure (T2F) and the Proportional Hazard Modeling (PHM) to obtain the optimal replacement time for the CNC machine tool [95]. The authors defined the tool replacement time at different machine settings using two types of analyses: tool availability and cost. The machining parameters were static settings, and it was adjusted before running the machine. Taha et.al [16] developed a self-healing mechanism for a CNC milling machine. This mechanism dealt with the CNC machine under fault and approved self-healing mechanism to the machine. The authors used pattern-recognition machine learning to define the recovery patterns and each pattern is bounded by corrective settings. The self-healing mechanism selects the recovery pattern

according to distance calculations to the current machine's faulty settings. From the selected pattern, the corrective actions are randomly selected through a uniform distribution that is bounded by the selected pattern zone. Table 7.1 provides a summary of many attempts in terms of what was achieved and what needs to be addressed.

Table 7.1 Research Gaps

Achieved	**Open Issues**
Offline optimal machining parameters	-Assuming ideal machining and environmental conditions -Static machining parameters
-Feed rate scheduling optimization -Adaptive control was applied to improve productivity	-Offline feed rate optimization -Complex modeling and empirical equations -Ideal tooling conditions
-Online optimization -Sensor predictions	-Massive online computation -Complex modeling -Ideal tooling conditions
-Tool wear monitoring -Failure alarm	-Passive system - Correction mechanism was not addressed
-Adaptive correction mechanism -Adding Tool wear monitoring to adaptive control	-Limited action values (feed rate) -Tool Time to Failure (T2F) was not addressed -Limited spindle speeds (Two values) -Discrete value adjustment
-Optimal Replacement. -Tool Time to Failure (T2F) -Different machining parameters	-Static Machining settings (non-variable) -Correction mechanism was not addressed -The failure detection module is time-independent
-Post-failure correction mechanism -Dynamic online machine settings. -Ease of implementation.	The study is missing the following: -T2F and degradation rate optimization -Optimal criteria to select the corrective settings -Effect of corrective action on productivity

To fill the research gaps presented in Table 7.1, the main objective of this paper is to develop an autonomous pre-failure mechanism that interacts with the CNC machine in the P-F interval to extend the tool's useful life. Figure 7.1 depicts the P-F curve, which is a conceptual curve of degradation of any physical assets. The P-F curve has two main points that express its name: the potential failure point (P), and the function failure point (F) [96]. Practically, the degradation process is a stochastic phenomenon, and each tool has a unique P-F curve [21, 23].

This section presents the LAD algorithm to generate the patterns, which are used to develop the self-healing actions in Module 3, and to classify and detect the out-of-specification in Module 2.

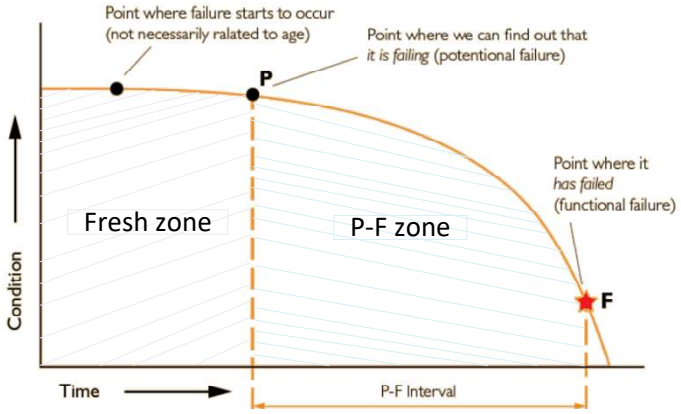


Figure 7.1 P-F Conceptual Curve

The proposed Pre-Failure approach has the following features:

1. Model-free adjustment mechanism for the CNC machine.
2. Continuous feed rate adjustment.
3. Time to failure extension and degradation rate slowdown.
4. Lower online computation efforts.
5. Applicable for wide ranges of machining parameters.

In this work, a model-free Deep Reinforcement Learning (DRL) is proposed for continuous online feed rate adjustment. This approach is developed to add a tuning mechanism that optimizes the tool performance and productivity in the P-F zone of the machine's tool. The approach has the capability to achieve the highest possible material removal rate while maintaining an acceptable tool wear level. This approach can be implemented in any machining process with less computational effort.
This paper is organized as follows: Section 2 describes the system layout and Pre-Failure mechanism procedure. Section 3 contains the physical experiment and data review. Section 4 presents the proposed methodology. Section 5 describes the results of the implementation and provides a discussion. Lastly, Section 6 concludes the paper.

7.2 System Description

7.2.1 System Layout

Figure 7.2 shows the system layout of the online autonomic closed loop for the CNC machine's pre-failure mechanism. In the pre-failure mechanism, there are two main phases and four modules. Phase 1 is the offline step for machine learning with Logical Analysis of Data (LAD), which is indicated by module 1. The software used is cbmLAD, which was developed for condition-based maintenance applications [32]. The cbmLAD is used to generate explanatory patterns that define the online P-F zones in module 2. The tool data is labeled according to the tool wear level VB as (a) new tool $VB < VB_p$, (b) Potential Failure (P-F) of the tool $VB_p < VB < VB_F$, and (c) Failure of the tool $VB < VB_F$. The data of the time-series forces are ingested into the cbmLAD to extract patterns that characterize the P-F zone. The tool's data is labeled as a failure when the tool wear level is more than, or equal to, a predefined value VB_F. The data in the potential failure zone is labeled in the same way, as shown in section 4.1.
P-F zone monitoring in module 2 is based on online rules extracted from the cbmLAD's generated patterns. This module monitors the tool performance and detects the instant of Potential failure and the tool failure. Module 3 represents the CNC machine in the Digital environment. The developed CNC Digital Twin (DT) is proposed to work online and in parallel with the physical machine. The DT model is supported by an artificial Neural Network (NN) that reads the CNC machine settings of cutting speeds speed (v) and feed rate f. It estimates the machine's forces measurements $[F_x, F_y, F_z]$ at each time step (t+1), based on the sensor's readings of forces at the time (t).

Module 4 is a Deep Reinforcement Learning (DRL) Pre-Failure agent that generates action a_{t+1} to adjust the feed rate f_{t+1} according to the optimal policy that the agent learned in the training phase. In the online mode, module 2 reads the CNC machine forces sensor measurements at time (t) and enables the Pre-Failure agent in Module 4 once the P-F zone is detected. The Pre-Failure agent reads the CNC machine measurement of the radial force F_x, the feed force F_y, the cutting F_z force, and the cutting speed v, at each time step. Accordingly, the proposed DRL agent adjusts the feed rate f_{t+1} to slow down the tool degradation rate.

7.2.2 Pre-Failure Intelligent Mechanism Procedure

This section presents the design steps to achieve the research objective of having a tool Pre-Failure mechanism for autonomous CNC machines. The proposed Pre-Failure mechanism has five main steps, as follows :

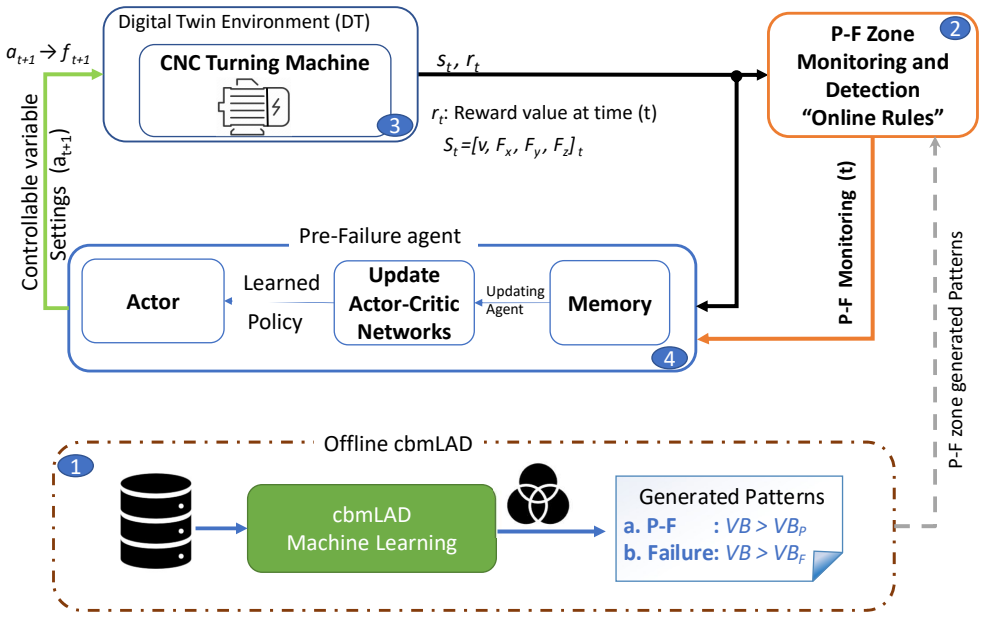


Figure 7.2 Autonomic closed loop to achieve Pre-failure Mechanism

1. CNC machine experimental data: the material-tool pair Time to Failure (T2F) data is an essential process in order to improve the tool performance in the Pre-Failure stage. In this paper, the raw T2F data is analyzed to monitor and detect the tool performance degradation in the P-F zone. The developed CNC machine's Digital Twin (DT) model is validated and tested with the Physical raw data. These data were collected for the process of turning Titanium Metal Matrix composite (TiMMC) material. The raw-data collection experiment was fully described in [56]. The data is presented in Section 3.

2. Tool P-F zone monitoring: the tool performance degradation is studied by building the tool P-F curve as the general one in figure 7.1. P-F curve shows the performance degradation versus the lifetime of the tool. Section 4.1 presents the data analysis of tool performance degradation and the proposed algorithm to define the tool potential failure instant. It also includes a Logical Analysis of Data (LAD) and the online generated patterns that monitor the P-F zone. By the end of section 4.1, Module 3 in figure 7.2 is achieved.

3. Deep Reinforcement Learning (DRL) Pre-Failure agent: This is the step of designing the DRL agent (module 4 in figure 7.2). Section 4.2 explains the DRL for continuous feed rate adjustment, and it defines the pre-failure agent objective and state vector that describes the CNC machine status from the perspective of DRL. Section 4.2 includes a description of the agent's architecture, training algorithm, and communication

links to modules 2 and 3. The added value and machining improvement for the tool's performance are discussed and verified in the results in Section 5.

4. CNC machine Digital Twin (DT): The DRL's environment is an essential part of Pre-Failure agent learning. A Digital Twin model is developed to interact with the DRL agent. The developed model lies on the collected experimental data, and it is validated with the physical machine tool's degradation. By the end of section 4.3, the CNC machine's digital module 3 in figure 7.2 is accomplished.

7.3 Review of the Experimental Data

The Proposed Pre-Failure algorithm will be implemented on a CNC machine during turning Titanium Metal Matrix Composites (TiMMC), and all experimental data used in this study is based on [56]. In the data collection phase, the experimental data was recorded under different static machining parameters on a 5-axis Boehringer NG 200 CNC turning center [56]. The tool diameter was 1.6 mm, and the tool wear was measured with an Olympus SZ-X12 microscope. In [56], two design variables are included; feed rate f (mm/rev) and cutting speed v (m/min). In terms of the machining outputs and experiment response, the forces and flank wear VB (mm) are recorded. The experiment consisted of five runs, and at least five replications for each run. A new tool was used for each replication. VB was measured every two minutes until its value exceed the failure level of 0.2 mm. The raw experimental data consists of 247 observations. A sample of the experimental data is given in Table 7.2. The observations indexed with (*) are the 1st observations of the tool's failure in each run-replication.

For example, figure 7.3 shows the radial forces of the experimental data provided in Table 7.2 for cutting speeds of 80 m/min. Figure 7.3 contains five replications of run 2 and run 4. It should be stated from figure 7.3 that increasing the feed rate at the same cutting speed leads to higher radial forces. Accordingly, the T2F becomes shorter when increasing the feed rate.

Table 7.2 A sample of experimental Raw Data

No	Run-Rep	t (s)	v *(m/min)*	f *(mm/rev)*	F_x (N)	F_y (N)	F_z (N)	VB (mm)
1	1-1	120	40	0.15	120.4	51.1	116.2	0.0525
2	1-1	240	40	0.15	126	50	109.4	0.06
:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:
59*	1-5	1560	40	0.15	452.1	74.5	162	0.2
60	2-1	20	80	0.15	85	32.7	83.2	0.035
61	2-1	40	80	0.15	86	34	84	0.0425
:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:
101*	2-5	270	80	0.15	780.9	138.6	191.7	0.3
102	3-1	30	40	0.35	113.1	42.8	171.5	0.04
103	3-2	60	40	0.35	118	46.7	176.7	0.0475
:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:
188*	3-6	1320	40	0.35	618.8	161.1	252.3	0.22
189	4-1	15	80	0.35	136.2	41.5	159	0.05
190	4-1	30	80	0.35	153	47	164.2	0.065
:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:
247*	4-6	120	80	0.35	880	224.8	293	0.25

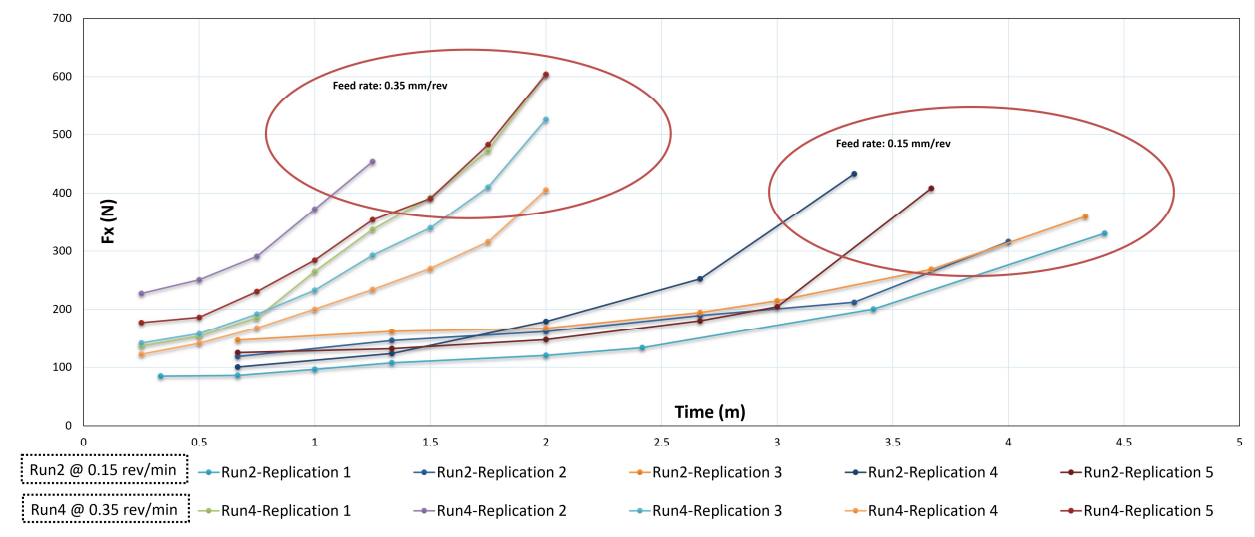


Figure 7.3 Experimental radial forces F_x at cutting speed of 80 m/min and different feed rates for different replications

7.4 Materials and Methods

7.4.1 Tool Degradation Monitoring on PF curve

Tool Potential Failure Point (P)

The P-F curve presents the tool performance's degradation against its operating time, and it defines the potential failure point (P) and the functional failure point (F) [96]. The tool has functionally failed when it exceeds the tool wear VB level that is recommended by the tool's manufacturer. The potential failure point (P) is the point at which the tool's failure propagation starts to increase, and it could be detected. In the P-F zone, the tool's performance has a significant deviation from its normal behavior when the tool is first installed. This performance is observed by the machine's sensor measurements of forces [56, 95]. P-F zone is an important mode as maintenance activities take place in this time interval [56, 95, 96].
In this paper, the knee point detection algorithm in [97] is adapted to define the potential failure point (P) for tool performance degradation according to the experimental tool wear data in Table 7.2. To plot the tool performance degradation P-F curve, an index that takes values in the interval of [0,1] is developed. The Normalized Tool performance degradation Index $NTPI$ is given by $NTPI = 1 - 5 \times VB$. It equals one with the new tool and zero at the tool failure limit of $VB = 0.2$mm. Figure 7.4 shows an example of the P-F curve for a tool operated at 40 m/min cutting speed and feed rate of 0.35 mm/rev. The potential failure point (P) is detected at 560 sec, and the wear is 0.073 mm for this replication.
The knee detection algorithm [97] calculates the Euclidian displacement between all of the points on the $NTPI$ graph and the perpendicular point on an imaginary reference line. This straight-line links the maximum and the minimum points of the tool performance, given by the dashed line in figure 7.5. The potential failure point is a point on the $NTPI$ that has the maximum positive Euclidian distance. In figure 7.5, the red curve is the Euclidian displacement between $NTPI$ and the reference line. Figure 7.6 depicts the Normalized Tool performance degradation Index $NTPI$ and the potential failure (P) points for all the runs and replications of the experimental data given in Table 7.2. The potential failure points are indicated by dashed lines. As the tool degradation is a stochastic process, the detected potential points are not the same for all of the runs and replications. Table 7.3 summarizes the potential failure levels of the tool wear VB for each run and replication for the cutting speed $v1 = 40m/min$ *and* $v2 = 80m/min$ and the feed rate $f1 = 0.15mm/rev$ *and* $f2 = 0.35mm/rev$. The average potential (P) tool wear over the collected experimental data is 0.135 mm. Tool P-F zone is the pre-failure zone at which the correction mechanism is needed.

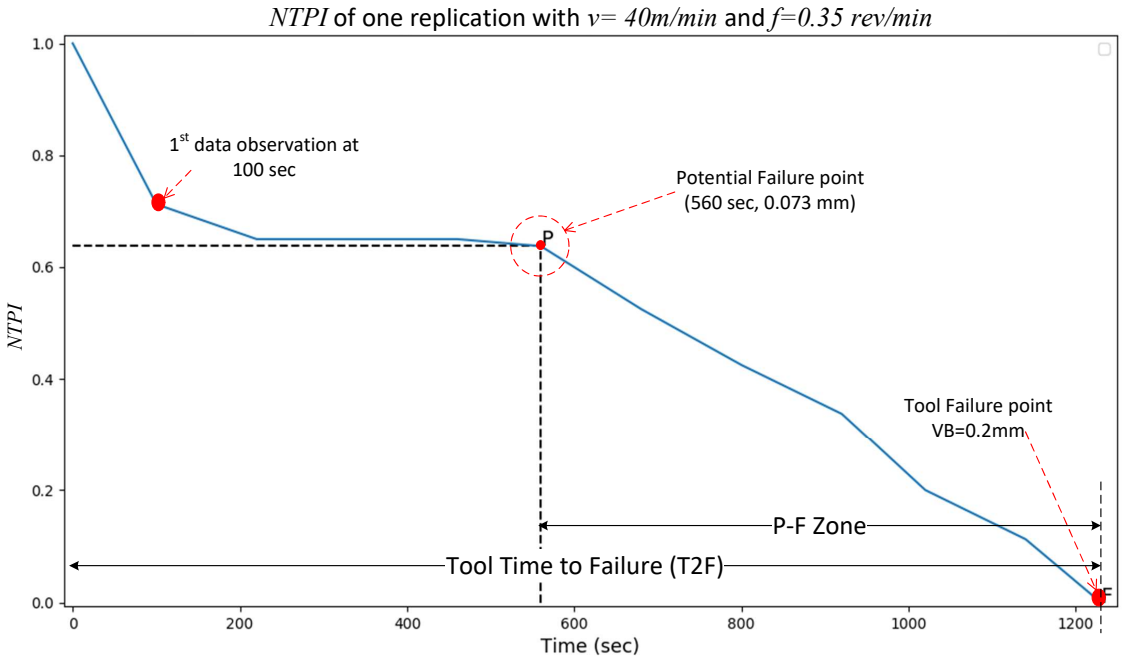


Figure 7.4 P-F curve of one tool replication under 40 m/min cutting speed and 0.35 mm/rev feed rate

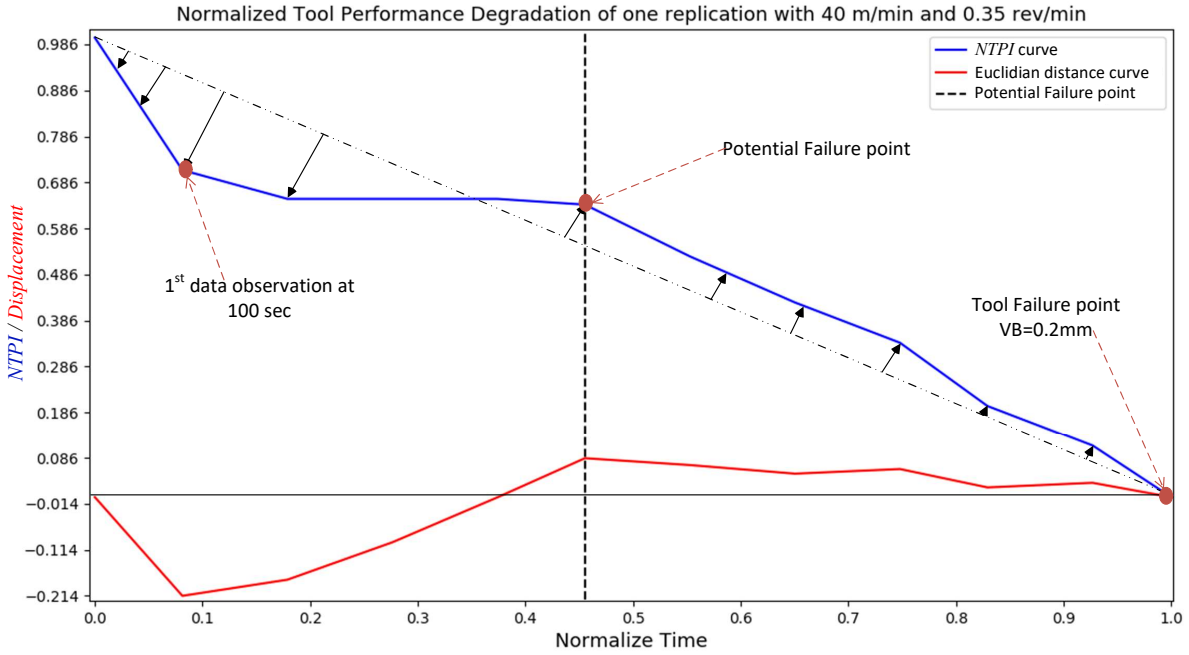


Figure 7.5 P-F curve and Euclidian distance curve for P-point detection

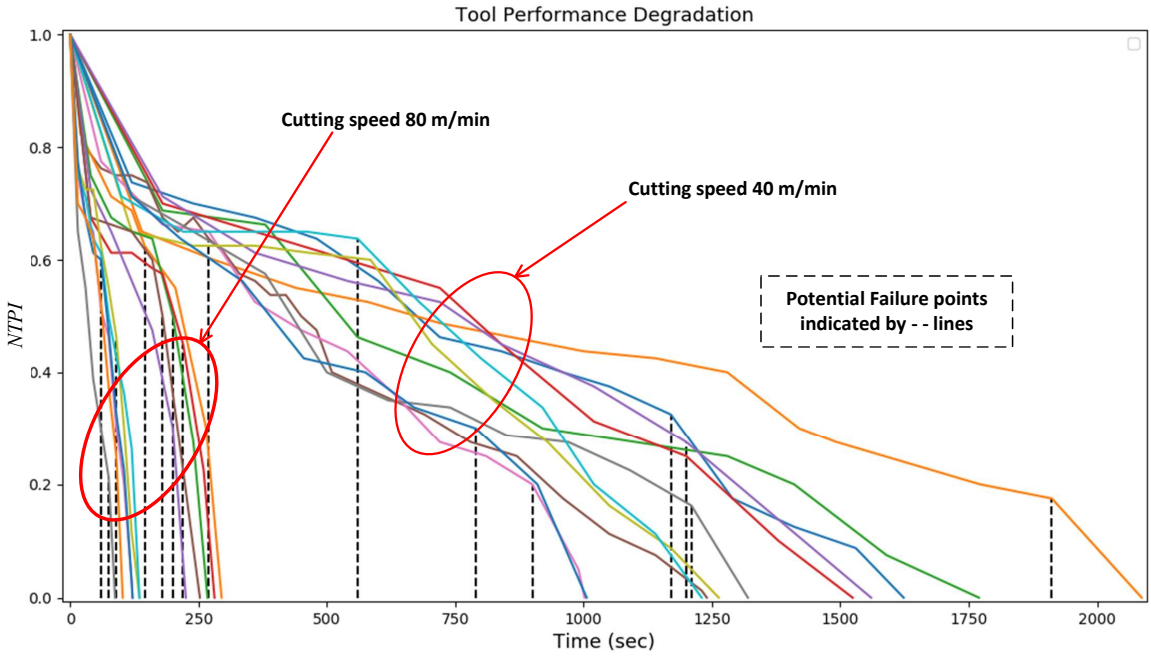


Figure 7.6 TiMMC Normalized Tool performance degradation Index $NTPI$ for different runs and replications of the experimental data in Table 7.2.

Table 7.3 Potential failure points of the CNC tool experimental data

***P*-tool wear (mm)**	**Rep 1**	**Rep 2**	**Rep 3**	**Rep 4**	**Rep 5**	**Rep 6**	**Average**
***Run 1 (v1*=40 m/min, *f1*=0.15 mm/rev)**	0.135	0.165	0.2	0.15	0.145	-	0.159
***Run 2 (v2*=80 m/min, *f1*=0.15 mm/rev)**	0.075	0.1	0.11	0.14	0.1	-	0.105
***Run 3 (v1*=40 m/min, *f2*=0.35 mm/rev)**	0.073	0.16	0.168	0.2	0.073	0.14	0.1357
***Run 4 (v2*=80 m/min, *f2*=0.35 mm/rev)**	0.2	0.158	0.108	0.077	0.2	0.095	0.1397
The average value of the potential tool failure							**0.135 mm**

Tool P-F zone online monitoring and detection

Logical Analysis of Data (LAD) is a non-statistical supervised data mining method. It uses Boolean logic functions and combinatorial optimization for classification [31, 32]. The advantage of LAD over other classification methods is to generate explanatory patterns for each class, which maintains comparative performance in knowledge extraction for supervised and semi-supervised classification problems. The patterns divide the multidimensional space of features into zones that characterize the classes.

cbmLAD solves Mixed-integer Programming (MILP) optimization problems iteratively to find the logical relationships among the input data features by generating patterns that characterize each class of the tool's life [28]. For each pattern, each feature is bounded by a specific range of values. For a new data point, the pattern is satisfied if the value of the measured features lies in its bounded range. In the one versus all (OVA) classification

technique, the cbmLAD generates patterns to characterize a specific class from the other classes. From the tool P-F curve, the tool-life consists of three zones, as shown in figure 7.7: (a) New-tool class, (b) Pre-failure class, and (c) Failure class. To detect the tool degradation state, cbmLAD divides the classification problem into three sub-problems and finds each class' discrimination function $\Delta 1, \Delta 2, \Delta 3$.

For a new observation O, the OVA cbmLAD multiclass's discrimination function $\Delta(O)$ is

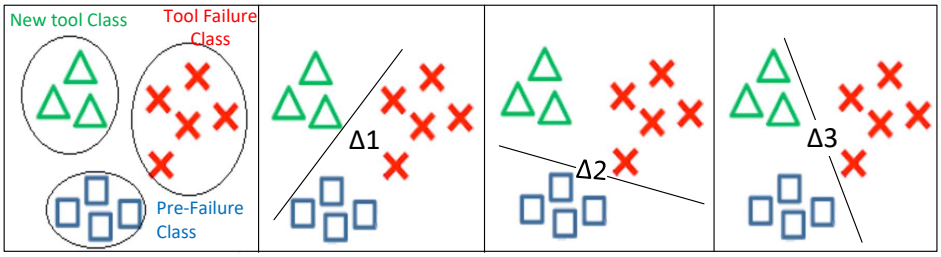


Figure 7.7 OVA Technique for the Tool Degradation Performance Classes in Two-Dimensional Space

given in Equation 7.1 as described in [28].

$$\begin{aligned} \Delta(O) &= arg_i max[\Delta_i(O)] \\ \Delta_i(O) &= \sum_{P_j} w_j \alpha_{P_j}(O), \ \forall i = 1, 2, 3 \end{aligned} \tag{7.1}$$

Where P_j is the j^{th} pattern that covers the observation O where j is the number of the pattern that belongs to the class i set of patterns. W_j is the pattern coverage weight, and α is a binary index, which is 1 when the observation O is covered by pattern P_j, and zero otherwise.

In this paper, the cbmLAD one versus all (OVA) technique is applied to solve a multi-classification problem in order to find the tool's state of degradation. In online mode at each time step (t), the P-F monitoring and detection module 2 in figure 7.2 monitors the time-stamped machine's sensors $[t, F_x, F_y, F_z]$ and checks whether the measured observation is covered by any of the patterns that represent the pre-failure or failure zones. The Pre-failure zone's detection signals are used to activate or deactivate the Reinforcement Learning RL module 4 in figure 7.2.

From Table 7.3, the potential failure VB level is 0.135 mm on average. The P-F zone is defined when the tool wear is $0.135 \leq VB < 0.2$, and the failure level is $VB > 0.2$. The experimental data is categorized into three main classes: (a) $VB < 0.135$, (b) $0.135 \leq VB < 0.2$, and (c) $VB \geq 0.2$ mm. According to the developed tool P-F curve, the data in Table 7.2 is identified

by the classes label and ingested to OVA cbmLAD to generate the tool life's patterns. Table 7.4 presents the generated patterns that characterize the P-F and the failure classes.
The online P-F monitoring module 2 in figure 7.2 performs two main functions, (a) the monitoring of the potential failure interval with ten patterns, and (b) the detection of failure with five patterns. Each pattern in Table 7.4 is represented in a multidimensional zone in the features space. At each time step (t), once the measured observation lies in a pattern zone, a signal is sent to activate or deactivate the Pre-Failure agent module 4 in figure 7.2. The P-F monitoring and detection module's scanning cycle is synchronized with the machine module 3 in figure 7.2 and with the pre-failure agent.

7.4.2 Deep Reinforcement Learning (DRL) Model

The standard RL is formalized as an agent that interacts with a system's environment, then receives the current system state and instant reward r_t at time t [33,34]. The RL goal is to find the optimal policy π^* that maximizes the return from the state $R_t = \sum_{t=0}^{\infty} \gamma r_t(s_t, a_t)$, where $\gamma \in [0, 1]$ is the discount factor for future rewards and t is instant of the return [29, 34]. The expected return value of taking action (a_t) in the state s_t under a policyπ is called action-value function or Q-function and it is equal to $Q^{\pi}(s_t, a_t) = E_{r_t, s_{t+1}}[r_t(s_t, a_t) + \gamma E_{a_{t+1}}[Q^{\pi}(s_{t+1}, a_{t+1})]]$. The optimal Q-value $Q^*(s_t, a_t) = max(Q^{\pi}(s_t, a_t))$ is the maximum returned value $\forall s_t \in S$ and $\forall a_t \in A$, where S is the state space, and the action space A is limited to discrete actions [29, 34]. The optimal policy is obtained from the optimal Q-value when obtaining the action that maximizes the returned Q-value; in mathematics, this is given by $\mu(s) = argmax_a Q(s_t, a_t)$ [29, 34]. The Q-learning is an off-policy algorithm that uses a greedy policy. It learns the optimal policy π^* as it approximates the Q-function by the Q-network parameters θ^Q. The optimal Q-function $Q^*(s_t, a_t)$ is achieved by obtaining the optimal parameters θ^{Q*} at which the training loss $L(\theta^Q) = E_{s_t, a_t, r_t}[(Q(s_t, a_t|\theta^Q) - y_t)^2]$ is the minimum. y_t is the target function with next time step state s_{t+1}, and it is calculated as $y_t = r(s_t, a_t) + \gamma Q(s_{t+1}|\theta^Q)$. For the Q-learning stability, the target y_t is calculated by another identical Q-network [34, 89]. Practically speaking, it is difficult to apply Q-learning to a continuous action space, and the Q-algorithm is not capable of optimizing an infinite number of actions at each time step. The actor-critic approach is used to solve this problem with the Deterministic Policy Gradient (DPG) algorithm [34, 89]. The critic is an action-value function $Q(s, a|\theta^Q)$ used to calculate the temporal difference (TD) error to criticize actions made by the actor, and it is updated based on the Q-function. The actor is a deterministic policy function $\mu(s|\theta^{\mu})$ that chooses action a_t given state s_t [34, 89]. The actor's network parameters θ^{μ} are updated according to maximizing the action-value $Q(s, a|\theta^Q)$, and its training losses are ascending losses given by $\nabla_{Q^{\mu}} J \approx E_{s_t}[\nabla_a . Q(s, a|\theta^Q)|_{s=s_t, a=\mu(s_t)} \nabla_{Q^{\mu}} \mu(s|\theta^{\mu})|_{s=s_t}]$. A Deep Deterministic policy gradient

Table 7.4 Generated patterns of P-F and failure zones for the data of the time-stamped Force

Class	***Pattern***	***t (sec)***	***Fx (N)***	***Fy (N)***	***Fz (N)***
P*re*-F*ailure*	1	$102.5 < t$	$305.3 < F_x < 685.95$	$126.85 < F_x < 155.85$	$203.3 < F_z$
	2	-	$305.3 < F_x < 542.3$	$117.25 < F_x < 174.5$	$F_z < 250.7$
	3	$85 < t$	$341.35 < F_x < 507.85$	$F_y < 174.5$	-
	4	$685 < t$	$271.2 < F_x < 605.4$	$71.3 < F_x < 174.5$	$175.25 < F_z < 250.7$
	5	$t < 67.5$	$354.1 < F_x < 507.85$	$93.75 < F_y < 174.5$	-
	6	$1085 < t$	$271.2 < F_x < 507.85$	$F_y < 174.5$	-
	7	$805 < t$	$261.9 < F_x < 507.85$	$78.25 < F_y < 114.6$	-
	8	$127.5 < t$	$316.1 < F_x < 507.85$	$F_y < 174.5$	$F_z < 190.25$
	9	$t > 685$	$249.5 < F_x < 488.5$	$71.3 < F_y < 174.5$	$139.7 < F_z < 159.85$
	10	-	$271.2 < F_x$	-	$143.75 < F_z < 150.2$
Class	**Pattern**	***t (sec)***	***Fx (N)***	***Fy (N)***	***Fz (N)***
Failure	1	$t < 327.5$	$F_x > 605.4$	-	-
	2	$1025 < t$	$F_x > 542.3$	-	-
	3	$t < 112.5$	$F_x > 543.3$	-	-
	4	$127.5 < t < 327.5$	$F_x > 488.5$	-	$F_z > 174.85$

(DDPG) is an algorithm that implements the deep Q-network on the DPG algorithm. DDPG approximates the Q-function and enables RL in systems that have continuous actions and a large-dimension state [34, 89]. DDPG is a model-free RL algorithm that uses a replay buffer memory to update the system's states, actions, and rewards during agent training [33,34]. In this paper, the pre-failure agent is an adapted DDPG algorithm to achieve optimal proactive and autonomous feed rate adjustment.

Pre-Failure Agent for Autonomous CNC Machine

In the current work, the CNC turning machine pre-failure agent action a_t is performed on the feed rate f_t mm/rev. At each time step t=1sec, the agent reads the machine sensor data $[F_x, F_y, F_z]$ and scans the P-F monitoring module signal. The agent generates actions to decrease the tool's degradation rate and keep a reasonable productivity limit in the P-F zone, which is given in Equation 7.2.

In the training phase, the pre-failure agent interacts with the DT model of the CNC turning machine, and it is rewarded for each action $a_t \rightarrow f_t$ with the reward function r_t. The rewarded value depends on the tool degradation rate and productivity at each time step. The machine's productivity is represented by the Material Removal Rate MRR (mm^3/min) in Equation 7.2.

$$MRR = f \times v \times d \tag{7.2}$$

Where f is the feed rate in mm/rev, v is the cutting speed in m/min, and d is the cutting depth in mm. The proposed pre-failure agent interacts with the CNC machine at a different cutting speed, which varies from 25 m/min to 80 m/min. The agent's actions are a wide

range of feed rate adjustments from 0.025 mm/rev to 0.35 mm/rev.

Definition of the State: The classical RL algorithm is an extension of a Markov Decision Process (MDP) and its assumption of time-independent states [29]. At each time step (t), the RL agent receives state s_t and takes an action $a_t \rightarrow f_t$ according to its learned policy [34]. To learn the optimal policy, the state is assumed to be time-independent, and it describes the system status regardless of the system's historical behavior. In many industrial system applications, the MDP cannot fully describe a system in which the time-independent state is useless in learning the optimal policy [34]. For example, in CNC machining, the instant sensor measurement $[F_x, F_y, F_z]$t at time (t) cannot abstract the tool wear stage, as described in section 4.1. Therefore, it is difficult to take a maintenance decision according to a time-independent instant value of forces. To ensure that the RL agent has the full features to describe the tool wear status, the RL state s_t is extended to include the cutting speed v (m/min), forces measurement at the instant of potential failure detection $[F_x, F_y, F_z]_p$, the sensor measurement deviation at time t from its value when the tool is at the P-point$[E_{xyz}]$, and the negative rate of forces $[\Delta F_{xyz}]_t$ over sampling time T. Pre-Failure agent's state s_t is given by Equations 7.3,9.1, and 9.2.

$$[E_{xyz}]_t = [F_x, F_y, F_z]_p - [F_x, F_y, F_z]_t \tag{7.3}$$

$$[\Delta F_{xyz}]_t = \frac{[F_x, F_y, F_z]_t - [F_x, F_y, F_z]_{t+1}}{T} \tag{7.4}$$

$$s_t = (v, [F_x, F_y, F_z]_p, [E_{xyz}]_t, [\Delta F_{xyz}]_t) \tag{7.5}$$

RL agent Action: The feed rate optimization is the key factor in optimizing the CNC machine tool's performance, as indicated in section 3. At the same cutting speed v (m/min), the tool's degradation rate decreases while the feed rate f (mm/rev) is decreased, and this decreases productivity. The pre-failure RL agent is designed to generate optimal and continuous action at that adjusts the CNC machine feed rate at each time step (t). This action aims at decreasing the tool degradation rate while keeping the productivity within an acceptable limit. In practice, the adjustable feed rate range depends on the tool-material pair, and for products in composite materials, it could be changed from 0.025 mm/rev to 0.35 mm/rev.

Reward Function: In RL, the reward function $r_t(s_t, f_t, s_{t+1})$ acts as the objective function in mathematical programming. At each time t, the RL agent explores the action space A to find the optimal action at that maximizes its reward according to the given state s_t. The pre-failure agent reward function is designed to minimize the tool degradation rate and to

keep the productivity level within acceptable limits. To maximize the tool Time to Failure (T2F), the Pre-Failure agent is designed with a positive reward function $r_t(s_t, f_t, s_{t+1})$ given by Equation 9.3. To keep the productivity level relatively high, the agent is rewarded only if its action at minimizes the forces deviation $|E_{xyz}|_t$,which is the absolute difference between the measurement forces $[F_{xyz}]$ in the P-F zone and the detected potential failure forces $[F_{xyz}]_p$ at $t = t_p$ point. The tool degradation rate increases when the deviation decreases. Equation 9.3 indicates the instant reward $r_t(s_t, f_t, s_{t+1})$ calculation at each time step t.

$$r_t(s_t, f_t, s_{t+1}) = \begin{cases} 1 & \text{if } |E_{xyz}|_{t+1} \leq |E_{xyz}|_t \\ 0 & o.w \end{cases} \tag{7.6}$$

Pre-Failure Agent Training

The Pre-Failure agent is an adapted DDPG algorithm, and its structure consists of the actor and the Q-function/Critic deep NNs. The actor adjusts the CNC machine feed rate according to the input RL state s_t and learned policy that is criticized by the Q-function network. To improve the DDPG agent training performance, a random noise $N_t \sim N(0, std)$ is added to the actor action [98]. The feed rate f_t to be adjusted at time (t) equals to $a_t = \mu(s_t|\theta^\mu) + N_t$, where $\mu(s_t|\theta^\mu)$ is the output of the actor-network and $\theta^\mu$is the parameters of the actor-network. In the training phase, the hidden layer's parameters of the Pre-Failure agent are updated to minimize the losses function of critics and to maximize the negative losses of the actor-network. To improve the learning stability, the target networks' parameters are updated with soft updates [98]; in other words, $\theta^{'} \leftarrow \tau\theta + (1-\tau)\theta^{'}$, where the learning rate τ is less than one.
The target network's parameters are $\theta^{Q'}$ and $\theta^{\mu'}$ for the critic and the actor. The pre-failure agent's full training algorithm is given in algorithm 1, and it is built and trained on a deep learning Pytorch environment. The developed Pre-Failure agent architecture and hyper-parameters are given in Appendix B.

7.4.3 Digital Twin (DT) for CNC Turning Machine

Recently, the development of Industrial IOT (IIOT), simulation modeling, and Artificial Intelligence (AI) enable the digitalization of the machines, and the Digital Twin (DT) is extracted as a new concept of Cyber-Physical Systems (CPS) [12, 23]. Digital Twin is a model that emulates the Physical CNC machine in the cyber/digital environment, and it has the capability of interacting with the real machine in the Physical environment [12–14]. DT was developed for a system level, and in the case of a single machine, DT is developed on

Algorithm 1 CNC machines' Pre-Failure agent training

Initialize Critic $Q(s, a|\theta^Q)$, Actor $\mu(s|\theta^\mu)$
Initialize target networks Q', μ'
for each episode: **do**
    Reads the cutting speed $\boldsymbol{v}$.
    for each K (min) steps in stimulation-Run: **do**
        scanning cycle time $t = k/T$
        switch Machine's tool P-F monitoring: **do**
            case New tool performance:
                Adjust the feed rate to maximum $f = f_{max}$
            case Tool in P-F zone:
                $[F_i]_P = [F_i]_{t_p} \quad , \quad \forall(i) \ \epsilon \ [x, y, z]$.
                Execute feed rate $f_t \leftarrow \mu(s_t|\theta^\mu) + \mathcal{N}_t$.
                Interact with CNC DT, get $[F_x, F_y, F_z]_{t+1}$.
                Calculate $s_{t+1} = (v, [F_{xyz}]_p, [E_{xyz}]_t, [\Delta F_{xyz}]_t)$
                $r(s_t, f_t, s_{t+1}) = \begin{cases} 0 & \text{if } |E_{xyz}|_{t+1} \leq |E_{xyz}|_t \\ 1 & o.w \end{cases}$
                Set targets $y_t = r_t + \gamma Q'(s_{t+1}, \mu'(s_{t+1}|\theta^{\mu'})|\theta^{Q'})$.
                Update Critic $L = \frac{1}{N}\sum_t (y_t - Q(s_t, a_t|\theta^Q))^2$.
                Update actor $\bigtriangledown_{Q^\mu} J \approx \frac{1}{N}\sum_t \bigtriangledown_a . Q(s, a|\theta^Q) \bigtriangledown_{Q^\mu} . \mu(s|\theta^\mu)$.
                Update $\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$.
                Update $\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$.
        if Tool worn **then** Break
        end if
    end for each
end for each

the component level. Figure 7.8 shows the implementation of DT on machine tool management. The digital environment contains physical data storage, data preprocessing, digital simulation models, and artificial intelligent agents. The digital environment has three main objectives: (1) monitoring the machine's data forces (2) analyzing this data to abstract the health status of the tool, and (3) taking action to improve the tool's performance. There are

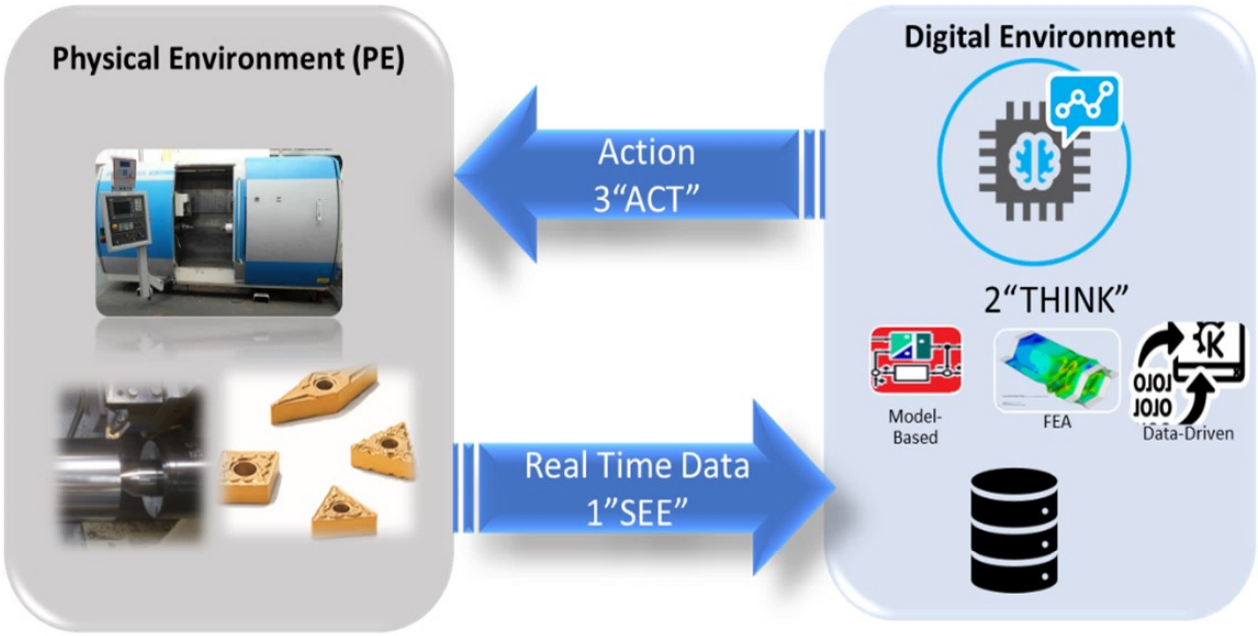


Figure 7.8 DT implementation on a machine's tool management

three methods to model a Digital Twin: Multiphysics modeling using Finite Element Analysis (FEA), Mathematical model-based, and/or Data-driven modeling [98]. In this paper, the experimental data in Table 7.2 is used to build the machine DT, and a deep artificial Neural Network (NN) is developed to act as a digital twin for the CNC turning machine. This model emulates the CNC turning machine in the digital environment. The DT's outputs are the estimated radial force F_x, feed force F_y, and cutting force F_z measurements at each time step t, and the model inputs are the cutting speed v (m/min) and the feed rate f (mm/rev), and time step t (min).

To minimize the overfitting of the model, a rule of sum for the NN architecture design is provided in Equation 9.4 [99, 100]. The numbers of hidden layers are limited to the number of inputs N_i and number of outputs N_o, while the number of hidden neurons N_h for N_s data observations is given by Equation 9.4 [99, 100]. β is a scaling factor that represents the

prevention of overfitting in the NN model and it takes a value from 2 to 10 [99, 100].

$$N_h = \frac{N_s}{\beta(N_i + N_o)} \tag{7.7}$$

The developed digital model has three inputs $[t, v, f]$ and three outputs F_x, F_y,andF_z. For 247 data observations, the number of hidden layers varies from 4 to 20 for $\beta \in [2, 10]$. One of the model's architectures is selected from more than 170 models' architectures. The models' hidden layers vary from single-layer to four-layer models, and each layer's amount of neurons changes from 4 to 20 neurons, and more five-layer models were added to the architecture's comparison. The best model is the model that has the lowest Mean Square Error (MSE) for the unseen testing data. Table 7.5 demonstrates the lowest MSE Network's architecture among all of the models with the same number of layers. The best model architecture of $[14 - 15 - 18 - 15]$ is selected. To build the CNC machine DT, this model is trained and tested on a deep TensorFlow learning environment [73]. During the testing of the DT model, the testing Mean Absolute Error (MAE) was ± 12.8 (e.g., Digital F_x = physical $F_x \pm 12.8$). Figure 7.9 shows the DT model sensors' reading detections versus the physical CNC turning machine sensors' reading for the unseen testing data of F_x, F_y, and F_z.

7.5 Analysis of the Results

This section analyzes the effects of the Pre-Failure agent on the tool performance and the tool Time to Failure (T2F) compared to the standalone CNC machine at different cutting speeds. The performance of the proposed Pre-Failure mechanism is measured by two key indexes: the Tool T2F and the achieved MRR. The P-F monitoring module activates the Pre-Failure agent in the P-F zone, and the agent is deactivated at the instant of tool failure. The closed-loop autonomy enables the Pre-Failure agent to adjust the optimal feed rate according to the estimated machine's forces at time (t+1). In online mode, the Pre-failure agent interacts with the CNC machine every 1 second, and its sampling time T is selected as T=60 sec.

The trained Pre-Failure agent is validated with the CNC Turing machine DT at different

Table 7.5 Lowest MSE NN models and their hidden layers and neurons

Layers	**1**	**2**		**3**			**4**				**5**				
Layer #	1^{st}	1^{st}	2^{nd}	1^{st}	2^{nd}	3^{rd}	1^{st}	2^{nd}	3^{rd}	4^{th}	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}
Neurons	14	5	19	4	6	19	14	15	18	15	4	10	16	10	14
MSE	**6.63E+04**	**6.44E+04**		**6.26E+04**			**5.53E+04**				**5.88E+04**				

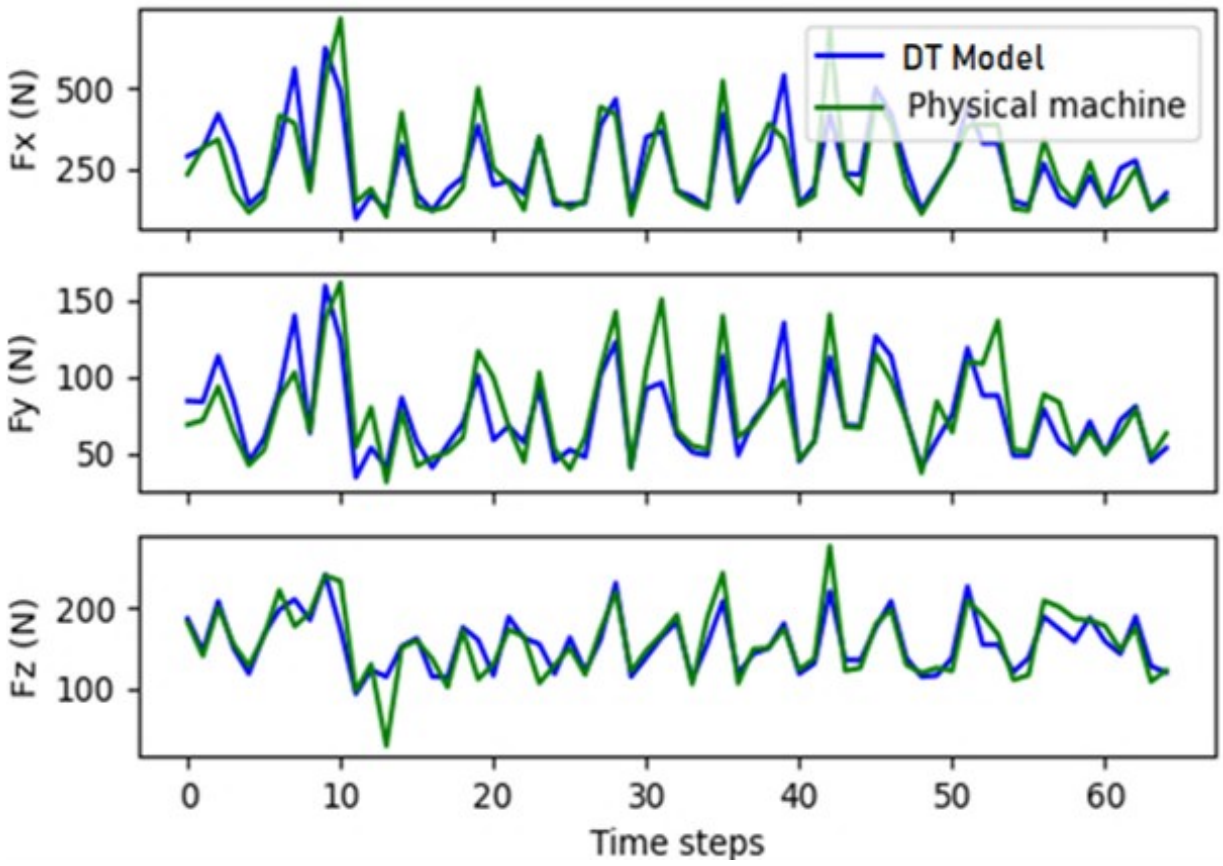


Figure 7.9 Sensor data of (a)F_x, (b)F_y, and (c)F_z with the CNC cyber model vs. experimental physical testing data

cutting speed settings given in Table 7.6. In each run, the autonomic CNC machine is simulated until the tool failure is detected at VB=0.2 mm. The machine's tool starts with a maximum feed rate of 0.35 mm/rev and the Pre-Failure enters the machining process when the P-F monitoring module 2 detects the potential-failure class and the instant of P point.

In Run I, the Pre-Failure agent increased the tool T2F by almost 27% over the standalone CNC machine. Figure 7.10 illustrates the force measurements for the standalone CNC machine in solid lines, and the tool Time to Failure (T2F) is 31.433 minutes (1886 seconds). With the implementation of the Pre-Failure agent on the CNC machine, the degradation rate

Table 7.6 Different speeds to validate the trained Pre-Failure agent

Simulation Run	**I**	**II**	**III**	**IV**	**V**
Spindle speed V(RPM)	5000	7500	10000	12500	15000
Cutting speed v(m/min)	25.12	37.68	50.24	62.8	75.36

of the machine forces decreases, which is indicated by the dashed $(- \cdot -)$ lines in figure 7.10. The tool's T2F increases to 40 minutes (2400 seconds).

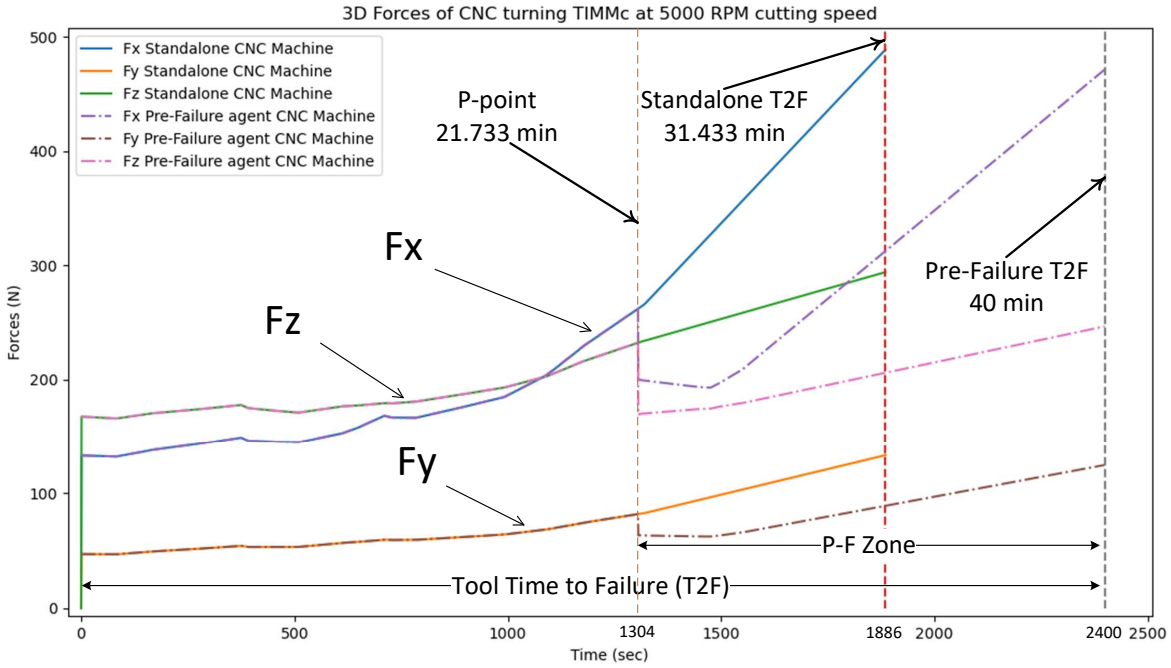


Figure 7.10 3D Forces F_{xyz} (N) at 5000 RPM of Run I for the standalone machine and the Pre-Failure machine.

In the P-F zone, the Pre-Failure agent generates a continuous feed rate adjustment according to the optimal trained policy in section 4.2. At a spindle speed of 5000 RPM, the adjusted feed rate at each time-step (1 second) is given with the blue color in figure 7.11, while its accumulated moving average is given in the orange color. At t=21.733 minute (1304 second,) the tool's potential failure point (P) is detected, then the feed rate is adjusted according to the learned optimal policy. Cumulative Moving Average (CMA) at each time step is plotted as the average of the Pre-Failure agent's action up to the current time-step. The Pre-Failure agent generates a variable feed rate at each second to maximize the T2F within the P-F zone.

The Pre-Failure agent keeps the productivity of the CNC machine at an acceptable limit, as the minimum CMA feed rate is kept at 0.2665 mm/rev. The machine's productivity index is the Material Removal Rate MRR (mm^3/min) in Equation 7.2. For a 0.2 mm depth of cut and 25120 mm/min cutting speed, the change of the MRR with the Pre-Failure agent is indicated by the green shaded area in figure 7.12. $MRR\%$ is the Pre-Failure agent's overall productivity MRR_{PF} relative to the standalone machine $MRR_{maxfeed}$ at the maximum feed rate given by Equation 9.5. $T2F_{st}$ is the standalone machine's T2F, and $T2F_{PF}$ is the T2F

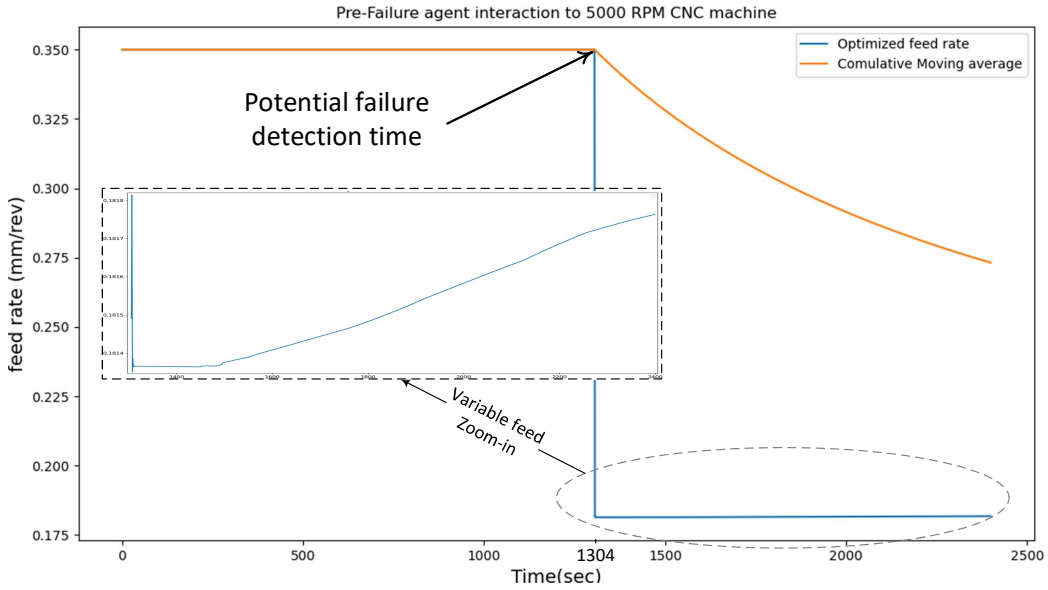


Figure 7.11 Pre-Failure feed rate adjustment at a spindle speed of 5000 RPM in Run I.

that includes the Pre-Failure agent.

$$MMR\% = \frac{\sum_{t=0}^{T2F_{PF}} MMR_{PF}}{\sum_{t=0}^{T2F_{st}} MRR_{maxfeed}} = \frac{\sum_{t=0}^{T2F_{PF}} f(t)}{f_{max} \times T2F_{st}} \tag{7.8}$$

In Run I, the Pre-Failure machine productivity is almost the same as the standalone machine with the maximum feed rate, and the $MRR\%$ equals 99.3%. The extension in T2F enables the machine to produce more within the added time. Figure 7.12 shows that the Pre-Failure added value MRR recovers the lost MRR.

Figure 7.13 concludes the extension of the tool's Time to Failure (T2F) by the Pre-Failure agent over the standalone machine. The tool's added lifetime is high with relatively low spindle speeds, and it is small with high speeds. The lowest T2F added time is 1.4 min (37%) for the tool that works on 12500 RPM and the highest added time is 10.9 min (50%) with 7500 RPM. At higher cutting speeds, the tool degrades faster, the P-F interval is smaller, and the Pre-Failure agent has a smaller time to interact with the CNC machine environment. In the meantime, the T2F added time adds more valuable MRR at high speeds, as stated in figure 7.14. The Pre-Failure agent keeps the level of productivity high, as given in figure 7.14, while the tool deviation from the potential failure level is considered as discussed in Section 4.2. The lowest Pre-Failure agent's $MRR\%$ is 79%, which is achieved at a spindle speed of 10000 RPM, as given in figure 7.14. At 10000 RPM, the tool T2F increases to 2.133 min (128 sec), and the added-value MRR with the Pre-Failure agent is lower than the lost one due to a decrease in the feed rate value. Figure 7.15 depicts the T2F and $MRR\%$

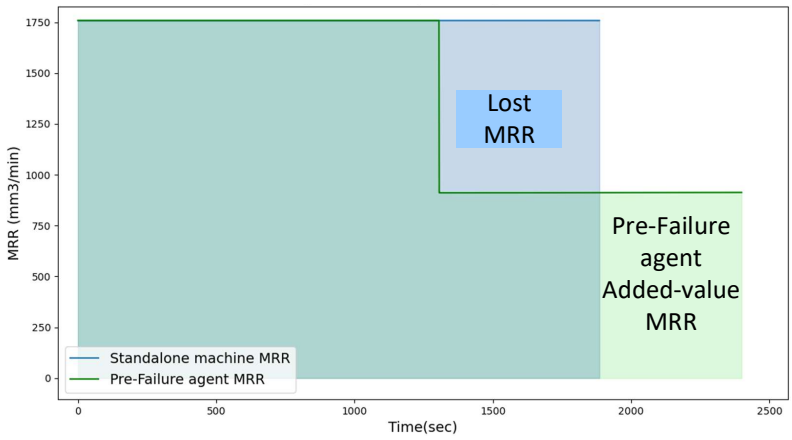


Figure 7.12 Pre-Failure Agent Tool MRR VS. Standalone Machine

of the standalone machine at different static feed rates and a spindle speed of 10000 RPM. The best standalone static feed-rate setting is 0.1745 mm/rev, which aims to increase both MRR and T2F; the achieved MRR is 59% of that at fmax, and tool T2F is 19.12 min. The proposed Pre-Failure mechanism outperforms the standalone machine with static settings, and the online agent's optimal policy achieves more T2F extension and $MRR\%$, as given in figure 7.13 and figure 7.14.

Implementation of the developed Pre-Failure agent improves the tool's performance in the P-F zone. The online optimal feed rate continuous adjustment adds on average of 5 minutes to the tool T2F and 5% to the MRR over the classical machining system. The detailed experimental results for each run in Table 7.6 are provided in Appendix C.

7.6 Conclusion

In this paper, the developed Pre-Failure approach improves the tool's performance in the Pre-Failure zone based on Deep Reinforcement Learning (DRL) during machining processes. The proposed Pre-Failure agent increases the tool Time to Failure (T2F) while maintaining the Material Removal Rate (MRR) at an acceptable limit. The machine tool's P-F curves and Logical Analysis of Data (LAD) are implemented to monitor and detect the potential failure level of the machine's tool. In the P-F zone, Pre-Failure model-free agent interacts with the CNC machine and adjusts its feed rate according to the estimated machine's forces at time (t+1). This method decreases the tool's degradation rate in the P-F zone before the tool is worn out, at VB=0.2mm. The Pre-Failure mechanism also keeps the forces at a relatively high level. To train the Pre-Failure agent, a machine Digital Twin (DT) was

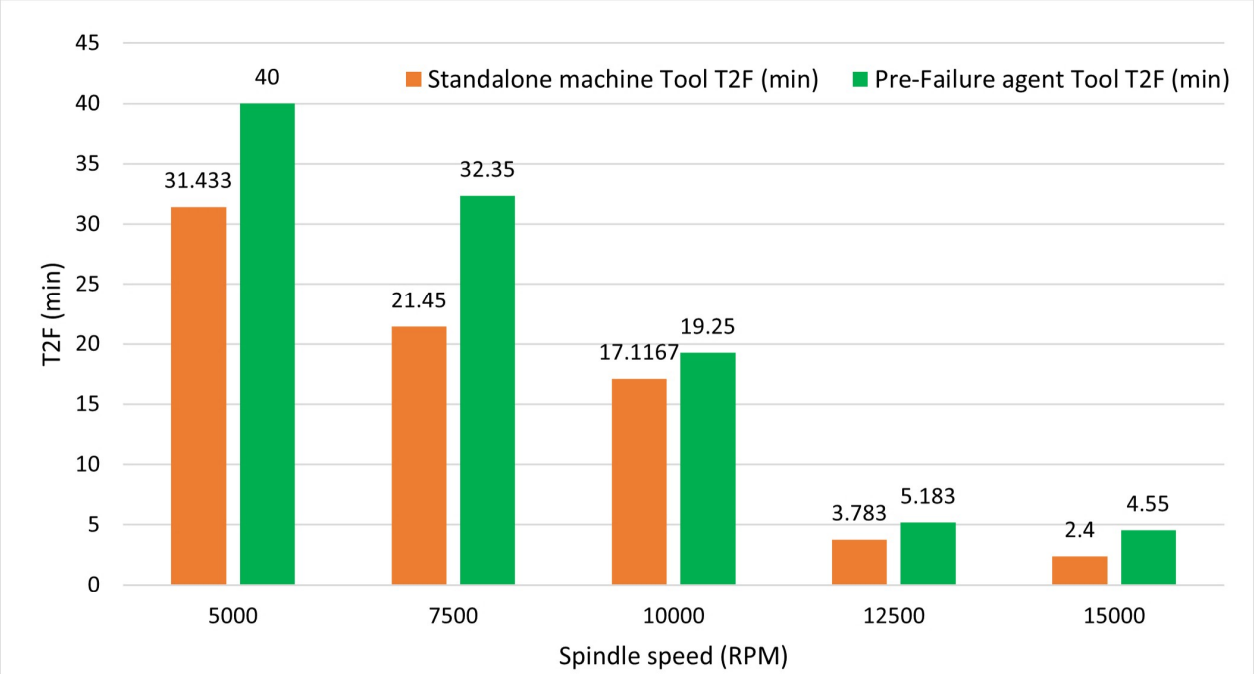


Figure 7.13 Tool T2F of the Pre-Failure and the Standalone machine for different RPMs

developed and validated with the physical machine data. The Pre-Failure agent is validated at different spindle speeds, starting from 5000 RPM to 15,000 RPM. By implementing the proposed Pre-Failure approach, the tool T2F increases over the classical machining approach. The value-added time is high at relatively low spindle speeds. It was found that the maximum added time is 10.9 minutes, which is achieved with 7500 RPM. Meanwhile, at 15,000 RPM, the tool T2F equals 4.55 min, which is almost double the standalone machine. In the P-F zone, the Pre-Failure agent adds more MRR that recovers the lost MRR due to decreasing the adjusted feed rate to be lower than its maximum value. At high speeds, the added MRR is higher than the lost ones. The Pre-Failure agent's MRR reaches 138.04% of that achieved with a static maximum feed rate under 15,000 RPM spindle speed. At 1000 RPM, the Pre-Failure agent gets the lowest MRRR of 79%, relative to the standalone machine, and it adds 12% of the tool T2F. However, the added time is not enough to recover the lost MRR. The developed dynamic Pre-Failure agent outperforms the best static adjustment for standalone machine runs at 10,000 RPM from the perspective of tool life and productivity. The current work can be extended in the future by including electrical power consumption, the material type, and other machining quality characteristics (e.g., surface roughness and residual stresses).

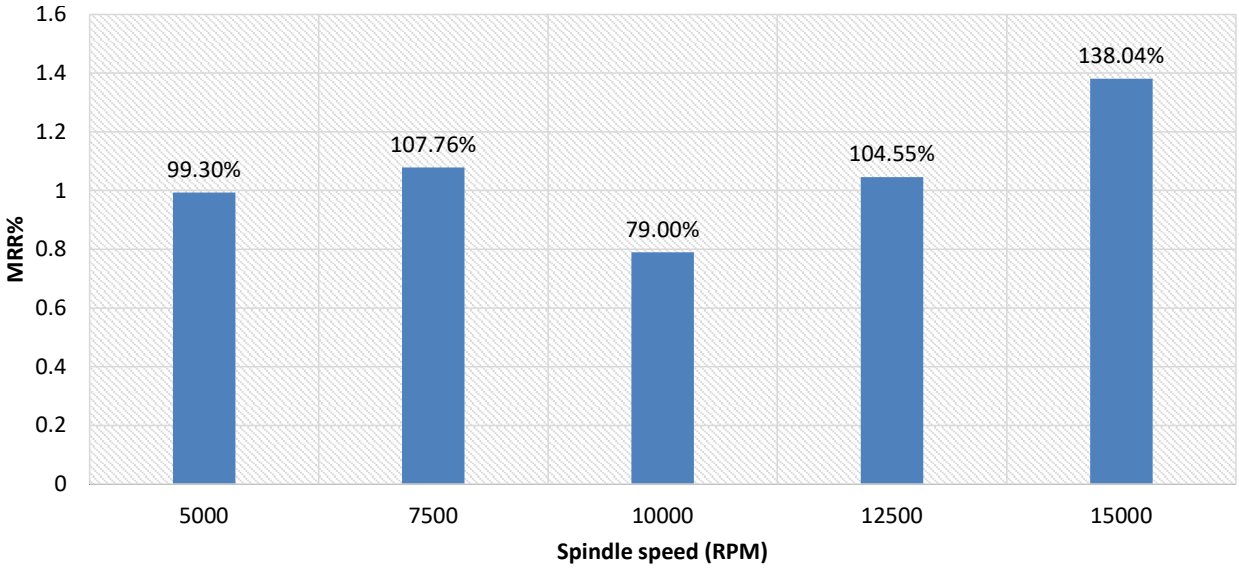


Figure 7.14 $MRR\%$ for different spindle speeds

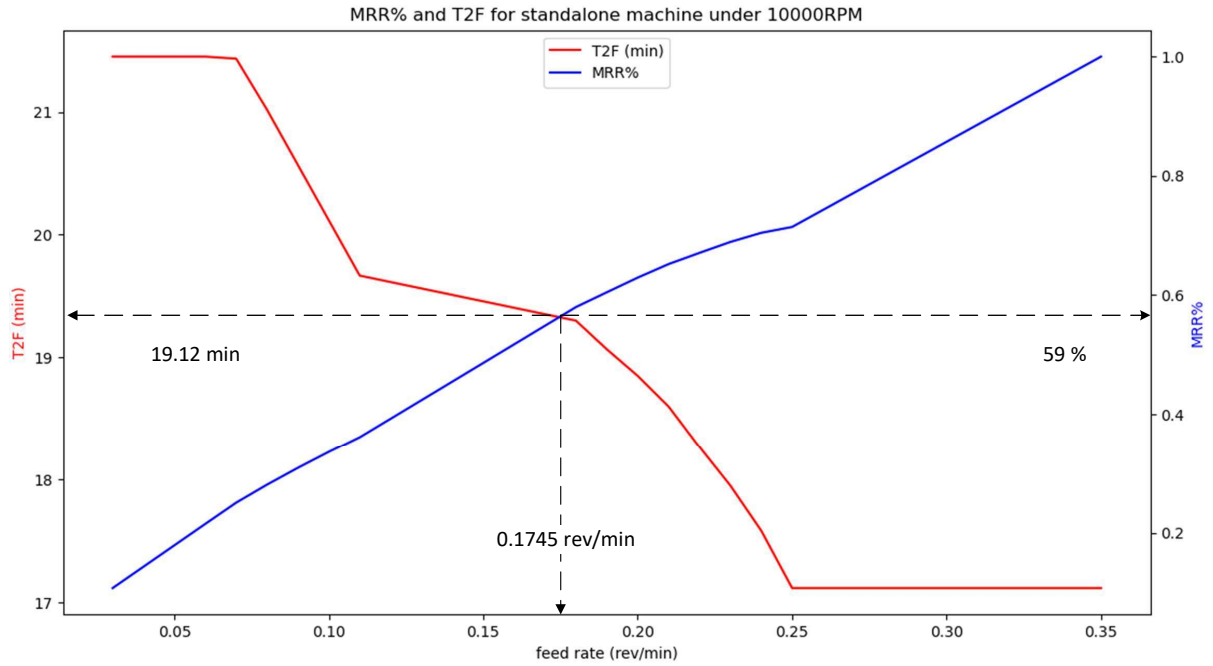


Figure 7.15 T2F and MRR% for a standalone machine at a spindle speed of 10000 RPM and different feed rates

CHAPTER 8 ARTICLE 5: FAILURE REASONING AND UNCERTAINTY ANALYSIS FOR WHEEL MOTOR ELECTRIC BUS

Hussein A. Taha, Ahmed H. Sakr, Soumaya Yacout, Phenicia Serafin
Published in: 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, 2021.

Abstract
Wheel Motor Electric Bus (W.M.E-Bus) is a recent e-mobility technology, which has a complex system integration. Since the operational reliability and life cycle data of such systems is scarce, it becomes impractical to plan for maintenance and determine system-critical components. Moreover, E-Bus system dismantling and assembling is a long time process especially for components near to the its power-system. In this paper, we propose a Fuzzy-logic fault-tree evaluation for the W.M.E-Bus system under uncertain failure data. The proposed method indicates the critical components that significantly influence the system's failure uncertainty. At a 10% failure rate uncertainty, control unit failure, including the embedded software, is ranked as the top critical failure mode with a 1.8 Fuzzy Importance Measure (FIM).

Keywords: Wheel Motor, Electric Bus, E-mobility, Fuzzy Logic, Fault Tree, Uncertainty, Fault Reasoning.

8.1 Introduction

Electrified Mobility (E-mobility) is the potential solution to reduce global warming, as it saves emissions and running costs. To achieve zero emissions by 2040, the public transportation buses are planned to be fully electrified buses (E-Buses) [101]. Failures in electric buses' driving system lead to road disasters [2, 102]. In these systems, parts replacement is a long-time process, in which power voltage has to be discharged before starting the replacement [103]. Since operational failure data are unavailable, reliability prediction and maintenance planning are challenging tasks.
Embedded software is critical in E-Buses as its failure is related to safety [102]. Hence, researchers address their failure in operation [4, 104]. Software Reliability Growth Models (SRGM) are utilized to model the embedded software failure. The fitted model goodness depends on the vehicle's operational modes and design [4, 104]. Pre-failure software testing

is proposed to eliminate operational embedded software failure [102].
Electrical motors are considered the main source of mechanical energy for the E-Bus wheels. Prognostics and Health Management (PHM) concepts are implemented to analyze the motor measurements such as current, temperature, and vibration [2, 105]. In addition to reliability analysis, DC/AC inverter fault-tolerant control techniques increase inverters' availability. PHM of electrical motors shows an accurate prediction of motor availability [106]. But, it addresses either the motor or the inverter individually. The unavailability of inverter affects the operation of electric motor and the E-Bus. Hence, researchers study the motor and its inverter as one system of a motor control system [2, 107]. However, the reliability analysis of motor control systems does neither address uncertainties of the failure data nor embedded control units. The main contributions of this paper are as follows:

- Pre-failure analysis of Wheel Motor Electric Bus (W.M.E-Bus) under uncertainty of failure data.
- Definition of W.M.E-Bus system critical components.
- Definition of the W.M.E-Bus's components that affect system's uncertainty the most.

The W.M.E-Bus system is described in section II. Section III introduces the proposed methodology. The obtained results and discussion are presented in section IV. Lastly, the conclusion is presented in section V.

8.2 system structure of Wheel Motor E-Bus

Wheel Motor E-Bus (W.M.E-Bus) system eliminates the differential unit between the two wheels [107]. The two motors' structure provides a symmetrical E-Bus's mass distribution [107]. The available W.M.E-Bus driving types are rear wheel drive, front wheel drive, or all wheel drive. In our studied system, the W.M.E-Bus consists of a driving axle and a passive axle. In figure 8.1, each wheel on the driving axle has a three Phase (3-ϕ) asynchronous electrical motor that is powered by a 3-ϕ DC/AC inverter. On the source side, battery pack supplies the DC power to the motor's inverter through DC link. The Red lines are hot voltage power lines. The green lines are the signal wires. The embedded control unit is a Digital Signal Processing (DSP) board that contains the W.M.E-Bus software. It provides the control signals to the 3-ϕ inverter, monitors battery's State of Charge (SoC), and performs the bus's driver inputs of braking or accelerating. For monitoring and controlling the bus's wheels, motor speed and temperature signals are sent to the inverter and to the embedded control unit.

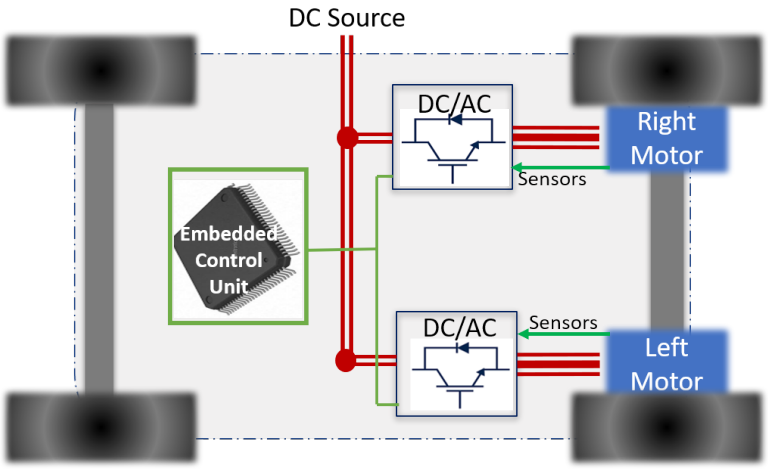


Figure 8.1 The Wheel Motor E-Bus System Structure.

8.2.1 Building Fault Tree

Fault Tree (FT) is a graphical and logical representation of potential failure modes, and it is implemented successfully in different applications [2]. It is proposed for pre-failure analysis and to identify the logical link between the W.M.E-Bus system and its basic components. Figure 8.2 shows the top three levels of the FT. The zeroth level is the top event of E-Bus system failure. The 1^{st} level of FT contains the main system items failure modes for the embedded control unit, and the two wheels. In the 2^{nd} level, middle failure causes of wheels' failure modes are indicated as the motor failure or the inverter failure. The reliability model

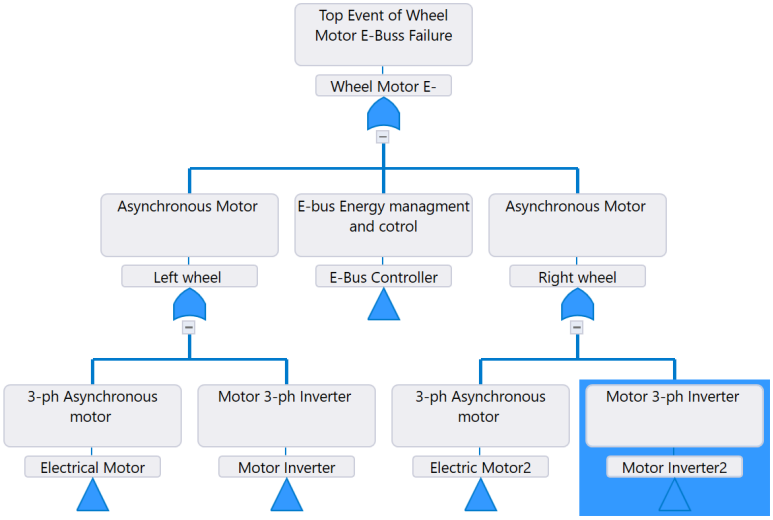


Figure 8.2 Fault Tree for The Wheel Motor E-Bus System Main Items.

for the embedded control unite is a series model. Its middle failure causes are given by figure 8.3. It indicates the basic components that have a high failure rate; main-PCB, main chipset, and the embedded software operational failures as stated in Table 8.1.

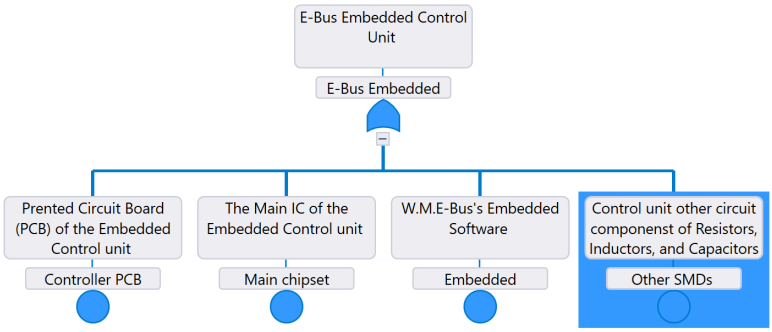


Figure 8.3 Sub-Tree for The W.M.E-Bus Embedded Controller Failure Mode.

Each wheel failure could be caused by either the 3-ϕ electrical motor or the inverter. The inverter failure mode is caused by either the inverter driving circuit, or sensing circuit. The frequently failed basic components of Inverter's driving circuit are Printed Circuit Board (PCB), power switches (IGBTs), and Isolation optocoupler (IC) as given in Table 8.1. The communication chipset, speed and temperature sensors are the basic events of the communication loop. Figure 8.4 depicts the 3-ϕ asynchronous motor's three main items: the stator, the rotor, and the bearing. These items are part of FT level 3. The bearing failure and winding failure are the frequently failed basic components of the electrical motor [1, 105]

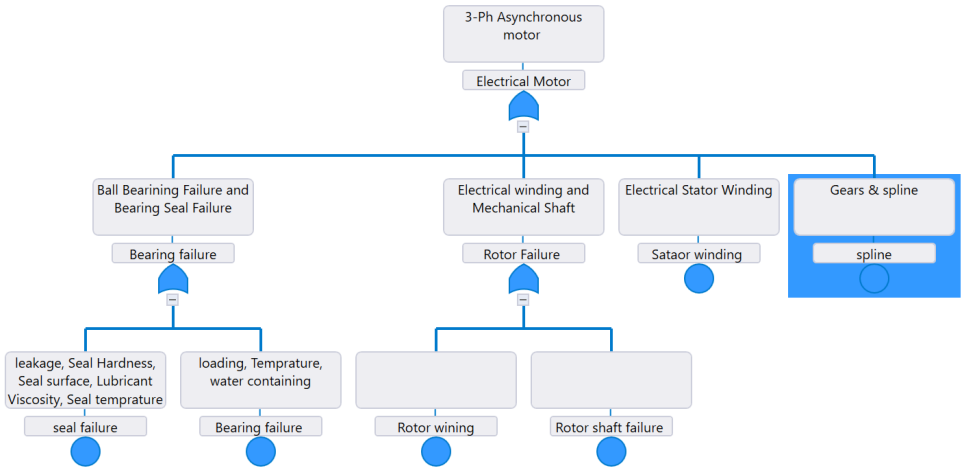


Figure 8.4 Sub-Tree for The E-Bus 3-ϕ Asynchronous Motor Basic Components.

Table 8.1 Wheel Motor E-Bus System Components and Failure Rates [1–4]

Component Description	**Symbol**	**Failure Rate/Mh**
Embedded Controller PCB	C1	3.06
Control unit Main chipset	C2	0.31
Controller Other SMDs	C3	0.072
Embedded software	C4	17.6
Left inverter PCB failure	L_i1	1.12
Left inverter IGBTs failure	L_i2	0.085
Left inverter Optocoupler failure	L_i3	0.243
Left inverter Other SMD's failure	L_i4	0.094
Left inverter communication IC failure	L_ic1	0.0252
Left motor temperature Sensor failure	L_ic2	0.22
Left motor Speed Sensor failure	L_ic3	0.0375
Left motor Bearing seal failure	L_m1	0.4465
Left motor Bearing failure	L_m2	0.083
Left motor rotor winding failure	L_m3	0.277
Left motor rotor shaft failure	L_m4	0.0226
Left motor stator winding failure	L_m5	0.277
Left motor Gear, spline failure	L_m6	0.0385
Right inverter PCB failure	R_i1	1.12
Right inverter IGBTs failure	R_i2	0.085
Right inverter Optocoupler failure	R_i3	0.243
Right inverter Other SMD's failure	R_i4	0.094
Right inverter communication IC failure	R_ic1	0.0252
Right motor temperature Sensor failure	R_ic2	0.22
Right motor Speed Sensor failure	R_ic3	0.0375
Right motor Bearing seal failure	R_m1	0.4465
Right motor Bearing failure	R_m2	0.083
Right motor rotor winding failure	R_m3	0.277
Right motor rotor shaft failure	R_m4	0.0226
Right motor stator winding failure	R_m5	0.277
Right motor Gear, spline failure	R_m6	0.0385

8.2.2 Fuzzy-Logic Analysis

The uncertainty in the W.M.E-Bus system's failure analysis is caused by a lack of operational failure data. Fuzzy logic fault tree analysis is a powerful method that analyzes the systems' failure probability under uncertainties [108]. Fuzzy logic analysis follows three main sequential steps [105, 108]:

1. Fuzzification.
2. Application of the Fuzzy logic rules.
3. Defuzzification.

Fuzzification is the step of converting the basic components' failure rate into fuzzy sets. For simplicity, we used a triangular fuzzy set, $[a_i, b_i, c_i]$, to represent the failure rate of component i. a_i is the failure rate's lower bound, and c_i is the upper bound. b_i is the middle value of the failure rate which equals the reference failure rate value λ_i. The membership function of this set, $\mu_i(r)$, is given by Eq. (8.1).

$$\mu_i(r) = \begin{cases} \frac{r-a_i}{b_i-a_i} & a_i < r \leq b_i \\ \frac{c_i-r}{c_i-b_i} & b_i < r < c_i \\ 0 & otherwise \end{cases} \quad , \quad \forall(i) \tag{8.1}$$

where r is the uncertain failure rate for each component, and i is the component index. The input fuzzy set's upper and lower bounds are calculated based on the uncertainty experts-defined Error Factor (EF). This factor represents the uncertainty of the W.M.E-Bus component failure rate as given by Eq. (8.2) [108].

$$a_i = \lambda_i(1 - EF) \quad , \quad c_i = \lambda_i(1 + EF) \quad , \quad \forall(i) \tag{8.2}$$

Assuming a constant failure rate λ_i for the electrical systems, the input fuzzy set failure probability matrix, $[F(t)]_{30\times3}$, is given by Eq. (8.3) [109] in which each row represents a single component. In our system, the number of components is 30. In Eq. (8.3) [109], t is the W.M.E-Bus system working hours.

$$[F(t)]_{i\times3} = [1 - e^{-a_i t}, 1 - e^{-b_i t}, 1 - e^{-c_i t}] \quad , \quad \forall(i) \tag{8.3}$$

The next step is the application of the Fuzzy logic rules according to the Fault Tree of the W.M.E-Bus system. The top event's reliability will be an exponential distribution, and its

parameter is the sum of the basic events' failure rate. Fuzzy logic rules are used to analyze the fault tree and to find the top event fuzzy set, $[F_0(t)]_{1\times3}$. According to our system's fault tree, as shown in figure 8.2, all relations are represented by OR logic gates. For N basic events in an OR logic relations, and a triangular fuzzy sets, the fuzzy probability array, $[F(t)_{OR}]_{1\times3}$, is calculated by Eq. (8.4) [108, 109].

$$[F(t)_{OR}]_{1\times3} = [1,1,1] - \Pi_{i=1}^{N}([1,1,1] - [F(t)]_{i\times3}), \tag{8.4}$$

Accordingly, $[F_0(t)]_{1\times3} = [F(t)_{OR}]_{1\times3}$. For the 30 components of our W.M.E-Bus, at $N = 30$, all the basic events are considered. $[F_0(t)]_{1,1}$, $[F_0(t)]_{1,2}$, and $[F_0(t)]_{1,3}$ are the lower, medium, and upper bounds of the top event fuzzy probability using a triangular membership function. Defuzzification is the last step of the fuzzy analysis. In this step, the top event fuzzy set probability, $[F_0(t)]_{1\times3}$, is converted into a deterministic value $TE_0(t)$. In the reliability context, the centroid approach is the most common defuzzification method [108], as presented by Eq. (8.5) for triangular fuzzy sets.

$$\begin{aligned} TE_0(t) &= \frac{1}{3}\sum_{k=1}^{3}[F_0(t)]_{1,k} \\ &= \frac{1}{3}([F_0(t)]_{1,1} + [F_0(t)]_{1,2} + [F_0(t)]_{1,3}) \end{aligned} \tag{8.5}$$

8.2.3 Fuzzy Uncertainty Analysis

There are two evaluation fuzzy factors to analyze the W.M.E-Bus system: the Fuzzy Uncertainty Importance Measure ($FUIM$) and the Fuzzy Importance Measure (FIM). $FUIM$ indicates the high impact basic components to the top event uncertainty. These components' data must be operational data to minimize the top event uncertainties [108]. At certain t working hours, $FUIM$ is calculated by Eq. (8.6) for component i. $[Q]_{1\times2}$ is the top event fuzzy $[lower, upper]$ probabilities. $[Q_i]_{1\times2}$ is the top event fuzzy $[lower, upper]$ probabilities when basic event i fuzzy set is a deterministic value with $EF = 0$. D_i is the Euclidean distance between $[Q]_{1\times2}$ and $[Q_i]_{1\times2}$, and is given by Eq.8.7.

$$FUIM = D_i([Q]_{1\times2}, [Q_i]_{1\times2}) \quad , \quad \forall(i) = [1, 30] \tag{8.6}$$

$$D_i = \sqrt{([F_0(t)]_{1,1} - [F_0(t)_i]_{1,1})^2 + ([F_0(t)]_{1,3} - [F_0(t)_i]_{1,3})^2}, \tag{8.7}$$

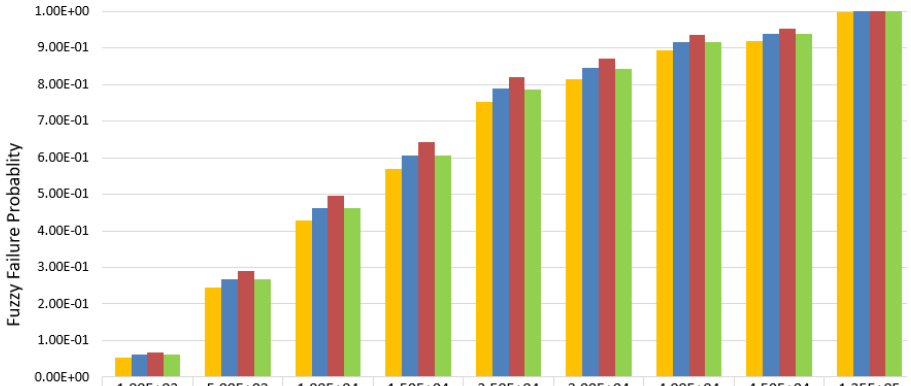
where $[F_0(t)_i]_{1,1}$ is the lower bound of the top event fuzzy probability when basic event i fuzzy set is a deterministic value with $EF = 0$. Similarly, $[F_0(t)_i]_{1,3}$ is the upper bound. We

did not consider the top event middle value in calculating evaluation factors, as the middle value of each basic components b_i does not contain the uncertainty error factorEF. The Fuzzy Importance Measure (FIM) identifies the critical W.M.E-Bus system's components, and it is given by Eq. (8.8). $[Q_i^1]_{1\times 2}$ is the top event fuzzy $[lower, upper]$ probabilities when basic event i is fully unavailable, i.e. $[F(t)]_{i\times 3} = [1, 1, 1]$. $[Q_i^0]_{1\times 2}$ is the top event fuzzy $[lower, upper]$ probabilities when basic event i is fully available, i.e $[F(t)]_{i\times 3} = [0, 0, 0]$.

$$FIM = D_i([Q_i^1]_{1\times 2}, [Q_i^0]_{1\times 2}) \quad , \quad \forall(i) = [1, 30] \tag{8.8}$$

8.3 Results

The reliability of the W.M.E-Bus system is evaluated under uncertainty using fault tree in figure 8.2 and Fuzzy logic reasoning with 10% uncertainty, $EF = 0.1$. Figure 8.5 represents the top event Fuzzy output sets, [Lower, Medium, Upper], at various working hours. The upper bound fuzzy output reaches the 0.8 failure probability before 25000 working hours. Therefore, the low-risk decision prioritizes the upper bound of the fuzzy set in the defuzzification step. The probability of failure for the top event reaches 0.8 at 62760 working hours.



Working hours	1.00E+03	5.00E+03	1.00E+04	1.50E+04	2.50E+04	3.00E+04	4.00E+04	4.50E+04	1.25E+05
Fuzzy LL	5.44E-02	2.44E-01	4.29E-01	5.68E-01	7.53E-01	8.13E-01	8.93E-01	9.19E-01	9.99E-01
Fuzzy medium	6.03E-02	2.67E-01	4.63E-01	6.06E-01	7.89E-01	8.45E-01	9.17E-01	9.39E-01	1.00E+00
Fuzzy UL	6.61E-02	2.90E-01	4.95E-01	6.42E-01	8.19E-01	8.71E-01	9.35E-01	9.54E-01	1.00E+00
Top_defuzzification	6.03E-02	2.67E-01	4.62E-01	6.05E-01	7.87E-01	8.43E-01	9.15E-01	9.37E-01	9.99E-01

Figure 8.5 Fuzzy probability Sets for different working hours.

Figure 8.6 shows the critical components of the W.M.E-Bus system at 10000 working hours and 10% uncertainty, $EF = 0.1$, according to Eq. 8.8. With a 1.82 FIM value, Embedded software failure is the most probable cause for W.M.E-Bus system failure. Moreover, the critical components are the Embedded controller's PCB (c1), Inverter's PCBs (L_i1&R_i1), and Motor seal Bearings (Lm1 & Rm1). Fuzzy uncertainties measurement (FUIM) factor is calculated by Eq. 8.6 for the W.M.E-Bus system under 10000 working hours. Embedded

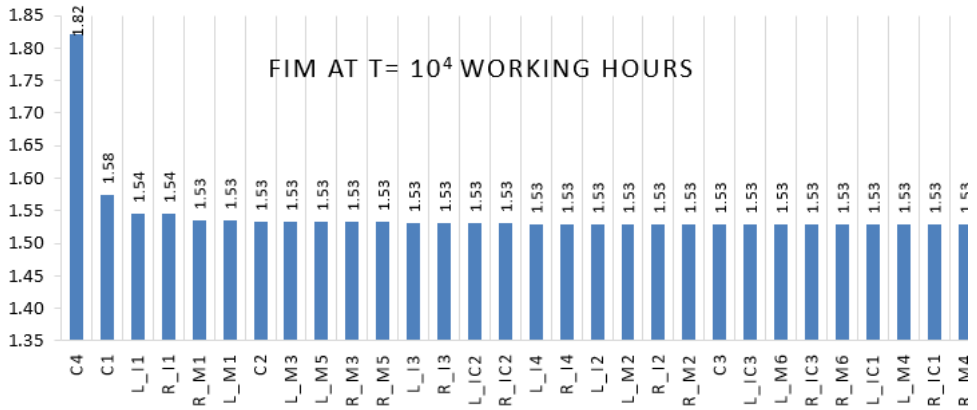


Figure 8.6 Fuzzy Importance Measure at 10000 Working Hours.

software failure uncertainty is the highest factor that affects the W.M.E-Bus system failure uncertainty and its FUIM is 4.9E-4. For the hardware level, the control unit' PCB is the highest uncertainty-influenced basic component with FUIM value around 1.2E-4. Uncertainties of the inverter PCBs' data are the 2^{nd} top basic components with 4.5E-4. The uncertainty of Motor bearing failure data due to the seal leakage comes in the 3^{rd} highest FUIM factor. These are the basic components that their data uncertainty significantly affects the W.M.E-Bus system's availability. The failure rates of these components are required to be well determined for more accurate system reliability analysis.

8.4 Conclusion

Wheel Motor Electric Bus (W.M.E-Bus) system structure is the potential heavy-duty electrical vehicle candidate as it minimizes the system cost and mechanical complexity. Since the system integrated E-Buses' failure data are unavailable, the maintenance and replacement suitable planning becomes a challenging task. We proposed a combination of fuzzy logic and fault tree to analyze the W.M.E-Bus system reliability under uncertainties of failure data. At 10% failure data uncertainty and 10000 working hours, we found that the embedded software was the most critical basic event of the Embedded control unit with an FIM index of 1.82. The 2^{nd} critical component was the embedded control unit's PCB with FIM value of 1.57 and FUIM value of 1.2E-4. The 3-ϕ inverter's PCB was the top second W.M.E-Bus critical component with FIM around 1.55 and FUIM value of 0.45E-4. Moreover, Electrical Motor important components were the bearing. In our future work, more W.M.E-Bus failure data will be provided, and reference failure data can be updated. This study will be extended to include the quality failures such as wiring connection failure. Downtime and cost will be included to have more applicable maintenance planning.

CHAPTER 9 ARTICLE 6: AUTONOMOUS UPTIME-IMPROVEMENT FOR ELECTRIC BUS IN DIGITAL TWIN ENVIRONMENT

Hussein A. Taha, Soumaya Yacout, Yasser Shabab

Submitted to: IEEE Transactions on Intelligent Transportation Systems.

Abstract

Autonomous and self-improvement systems rely on the historical operational data of these systems. With the lake of operational data, the development of the self-improvement process becomes more challenging. This paper develops an autonomous uptime improvement scheme for a system with either unavailability or uncertainty of operational failure data. Fuzzy-Fault Tree Analysis (F-FTA) is adapted to analyze the system failure with operational data uncertainty. Meanwhile, the Reliability Centered Maintenance (RCM) worksheets are developed to define the uptime-improvement autonomous actions. The proposed scheme is verified on the Wheel Motor Electric Bus (W.M.E-Bus) drive system in the Digital Twin (DT) environment. It links the on-road Physical W.M.E-Bus driving conditions to the system failure and reliability estimation, which are in the digital DT environment. The Embedded Control Unit (ECU) transmits the W.M.E-Bus's driving cycle to be analyzed in the digital environment. Then, the improvement action of derating percentage for the driving cycle is sent back to the E-bus to automatically manipulate the driver's pedal in the next driving cycle. In this particular case study, the deployment of the proposed scheme increases the maximum allowable W.M.E-Bus millage distance by 8603 miles when it reduces the driving cycle by 36 %, on average. A detailed improvement analysis for different operating speeds and RCM worksheets are reported.

Keywords: Electric-Bus, Reliability, uptime, Digital Twin, Data uncertainty, RCM, Derating, E-mobility.

9.1 Introduction

Electrified Mobility (E-mobility) is a potential solution to tackle global warming, as it saves emissions and running costs. Global EV outlook is an annual publication that identifies and discusses the recent developments in electric mobility. Their report of 2021 stated that many countries have planned to have a fully electrified transportation by 2040 [110, 111]. This massive transient makes the maintainability and the uptime of the Eclectic-Bus an essential requirement, especially for the complex integrated-system buses [111]. The part replacement

of a Wheel Motor Electric-Bus (W.M.E-Bus) is a long-time process, where the high voltage has to be discharged before starting the replacement [110]. The up-time extension activities for this system are highly required by fleet management.

The combinations of Industrial 4.0 and Artificial Intelligence enable real-time monitoring and increase the uptime of the E-Bus [16, 23]. The implementation of these new strategies is highly dependent on the accessibility of the historical operational data [17]. For the in-service E-Buses, the full system connectivity and the run-to-failure data are still unavailable. This paper fills this gap and provides the necessary structure for autonomous up-time improvement while the system operational data are unavailable.The literature shows attempts to address the E-Bus system failure analysis, and maintenance [2, 110, 112]. Reliability Centered Maintenance (RCM) is a concept that was introduced to reduce failures specially those caused by inadequate maintenance [30]. RCM was adopted in the passenger vehicle to avoid system failures while creating a periodic replacement of the system's components [112]. To get the system's state of failure, the system failure analysis must be added. Shu et.al 2019 applied the Fault-Tree approach to study the reliability of a central-drive electric vehicle [2]. The study assumed a certain failure data and the drive system was analyzed in the design stage. Taha et.al 2020 [110] studied the failure criticality analysis for the W.M.E-Bus drive system in the operating stage considering data uncertainty. These studies are passive approaches, and neither the real operating condition of W.M.E-Bus nor self-improvement activities were addressed. To fill this gap, this paper present a proactive approach that analyses the real operating condition of the E-Bus in case of unavailability of data and provides uptime improvement's actions.

Fault-tolerant control (FTC) was developed to increase system availability and reduce the risk of safety hazards. FTC is intended to compensate for the fault effects on the system during operation to maintain the system stability. To react to system malfunctions, FTC includes online fault detection, automatic condition assessment, and the calculation of remedial actions when a fault is detected [113–116]. FTC is designed to recover the vehicular drive system after having a pre-specified component failure, for example a speed sensor failure [113, 115], temperature sensor failure [116], or inverter switch failure [114]. FTC is a post-failure reactive approach where it reacts to the system after a failure incident. Samaranayake et.al 2018 developed a proactive Model Predictive controller (MPC) to address the electrical motor degradation [117]. Firstly, MPC estimates the motor power losses that represented the motor degradation. MPC controls the supplied voltage to decrease the internal losses [117]. Both FTC and MPC addressed a dedicated system component, and the system integration is missed. To fills this gap, this paper develops a scheme to improve the reliability of the W.M.E-Bus integrated drive system.

Table 9.1 summarizes the successive research to improve the reliability of E-Bus and the research gap that needs to be addressed. The proposed scheme in this paper fills those gaps by presenting autonomous uptime-improvement for W.M.E-Bus systems with unavailable operational data. This scheme links the on-road real driving condition of the W.M.E-Bus to the system analysis and the proactive actions to increase the system up-time. The developed

Table 9.1 Summary of the W.M.E-Bus Drive System uptime Research Gaps

Achieved	**Need to be addressed**
-Drive system analysis -Combination of design failure data	-System analysis in operation mode -System failure time and replacement time
-Power converter failure analysis under data uncertainty	- Integrated System failure analysis -Autonomous active approach
-Drive system failure analysis -Failure data uncertainty -System critical components definition	-Passive approaches -A real operating condition -autonomous corrections
-RCM Tasks worksheets.	-Passive planning -Integrated-System Failure analysis
-Reaction to the component failure. -Tolerate the component's failure	-Proactive approaches. -System-integration needs to be addressed
-Proactive Motor degradation control -Degradation improvement of electrical motor	- Integrated System Failure analysis and autonomous actions -Real on-road operating condition.

scheme establishes the self-improvement activities of the W.M.E-Bus drive system in the Digital Twin (DT) environment, and adds the following features:

- Analysis of the system failure while its operational data are uncertain or unavailable.
- Definition of the autonomous actions that improve the uptime of the integrated E-Bus system.
- Proactive approach to extend the E-Bus uptime.
- On-road real operating conditions analysis of E-Bus and application of automatic actions to increase the bus allowable mileage.

This paper is organized as follows, Section 2 describes the Proposed scheme layout and its main modules.Section 3 contains the methodologies that are proposed to develop a self-improving system which lakes operational failure data. Section 4 shows the implementations and verifications of the W.M.E-Bus. Section 5 presents the achieved results. Section 6 is the conclusion and future work.

9.2 Autonomous uptime-Improvement Scheme Layout in DT

The concept of autonomous available systems relays on two main functions: monitoring the system health and performing improvement actions [16, 17]. The autonomic loop is a closed-loop that links these two essential functions of autonomy and enables the autonomous action to be executed [16, 17]. The proposed implementation scheme of the autonomous uptime extension in the Digital Twin (DT) is given in Figure 9.1. The digital environment contains three modules as follows:

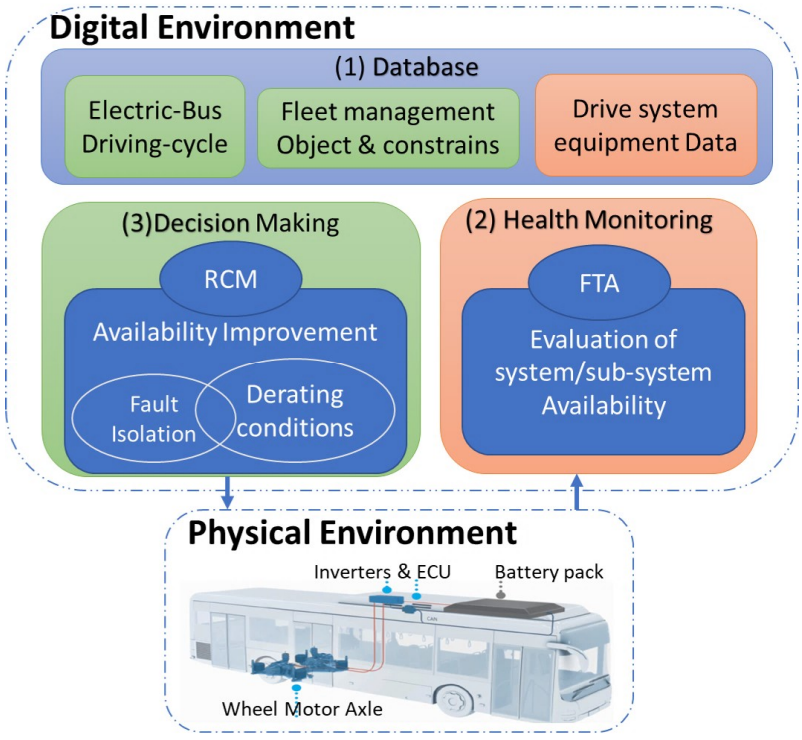


Figure 9.1 Autonomous uptime-Improvement Scheme for W.M.E-Bus In DT.

1. Database

   The database is the historical data of the E-Bus system. These module includes the fleet management data such as the maximum allowable mileage before performing maintenance process, the scheduled time for each bus-stops, and the realtime driving condition data [17, 118]. In case of unavailability of the historical failure data, similar failure data for the basic components of the system is involved with an uncertainty percentage [110, 119]. To obtain the similar failure data, the common references are the IEEE golden book for power system components [120], USA military handbook for electronic

components [121], and US Navy reliability handbook for mechanical components [1].

2. System Health Monitoring.
   In the health monitoring module (2), the conditional monitoring and diagnosis of the system is performed [16]. In this module, the system failure is analyzed to evaluate the health state of the system and its subsystems. In the application of this paper, the health monitoring module is connected to the Database module (1) and to the physical W.M.E-Bus to estimate system reliability at each time t. As the operation historical data of the studied system are unavailable, the FTA analysis with failure data uncertainty is proposed to Analyze the system failure-probability and reliability at a time t, as explained in section 3.1.

3. Decision Making.
   According to the analysis of the system reliability in module 2, the decision module provides autonomous actions to improve system uptime. These actions are extracted based the RCM's worksheets to enable the autonomic closed-loop between Physical and digital environments [109], as described in section 3.2. In the W.M.E-Bus, these actions include changing of the driving conditions automatically, and isolating of the failed components, as stated in section 4.2. The proposed scheme of the uptime improvement is verified on a W.M.E-Bus drive system while monitoring and improving the driving cycle autonomously.

9.3 Methodology

Reliability and maintenance are built on efficient definition and diagnosis of faults/anomalies and failures. They address the optimal actions of maintenance or replacement time [30, 109]. Reliability Centered Maintenance (RCM) is a concept that was introduced to reduce failures caused by inadequate maintenance for several industries [30]. Maintenance is applied to ensure that the machine continues to perform as intended, while RCM addresses modes of failure and their consequences and possible maintenance actions. These actions are chosen to improve the maintenance function and minimize the influence of failures because of inadequate decisions. The RCM approach identifies the potential causes of failure, failure effects and consequences, and possible actions to prevent or reduce the risk of failure [30]. There are several tools to apply the RCM concept namely: Failure Mode and Effect Analysis (FMEA), Consequence of Failure Analysis (COFA), and the Top-down approach of Fault-Tree Analy-

sis (FTA) [30, 109]. To avoid a failure mode, RCM defines the proper tasks/activities to be taken. RCM concept follows two main phases [30]: (1) Identifying the system components and subsystems critical failure mode, (2) Specifying the RCM actions for each failure mode. The 1^{st} phase addresses the system and subsystem failure analysis, defines the possible causes of the system failure, and estimate the system reliability at a time t. The system/subsystems design has to satisfy the required safety and environment conditions. For example, in Wheel Motor Electric-bus (W.M.E-bus), passenger safety is a requirement, and the reliability of the bus is an operating criterion. In this paper, the Fault-tree is used to define and analyze the W.M.E-Bus system/subsystems' failure, causes, and estimate the E-Bus's reliability at a time t, as demonstrated by section 3.1.
The 2^{nd} phase develops RCM worksheet that indicates the actions to be taken to mitigate the causes of each failure mode. The RCM actions are categorized into pre-failure and post-failure actions [30, 109]. This paper addresses the Pre-failure RCM actions that could be autonomously executed to increase the W.M.E-Bus system's uptime. Section 3.2 describes the RCM's actions selection criteria and the RCM's worksheet structure.

9.3.1 System Health Monitoring and Failure analysis

As the in-field operational failure data of the W.M.E-Bus system are unavailable, this paper develops an autonomous uptime-improvement scheme based on data uncertainty. The system failure probability $F_i(t)$ at time t is extracted by the Fuzzy Fault-Tree Analysis (FFTA) algorithm for data uncertainty [110]. The FFTA algorithm has four main steps, as follows [110]:

1. Building the system Fault-Tree and defining the basic components of each subsystem i.

   Fault-Tree is a top-down logical approach that presents the potential failure modes for a system. The tree starts with the main system failure at the top level, then moves down until reaching the root causes or basic events level. The middle levels represent the subsystems, and there are logical combinations of intermediate events that lead to system failure [110]. Figure 9.2 demonstrates a simple circuit example of two parallel motors and its Fault-Tree. In this example, motors are connected to the DC source through a switch. The main failure mode here is No-Rotation, which is indicated on the top level of the tree. The intermediate events or the subsystems give the possible causes for this failure mode, which are the failure of motors subsystem or power supply failure. The basic event indicates the subsystem's basic components, which are the root causes that could lead to the main failure mode. In case of a lack of operational failure

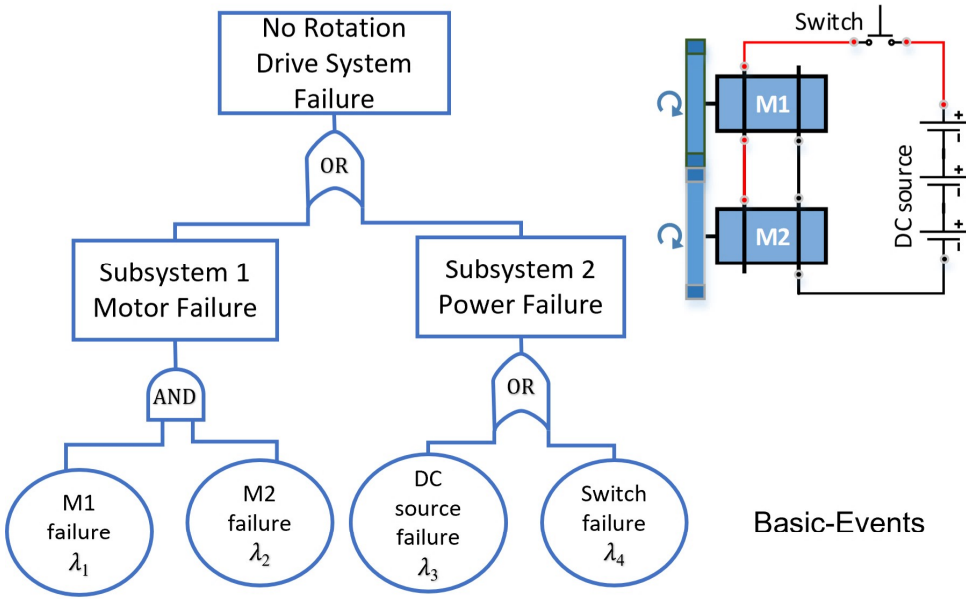


Figure 9.2 Fault Tree Example For No-Rotation Failure Mode In Simple Dual Motor circuit

data, the failure rate for each basic component is obtained based on similar reference data as uncertain data. the proposed FFTA is a successive algorithm to analyze the system's failures under data uncertainty [110, 122, 123].

2. Fuzzification.

   For basic component k, the uncertainty of the failure rate is represented by a fuzzy set with a mathematical distribution. The uncertainty factor Ef_k for the k^{th} component is a factor that is used to define the fuzzy sets, and its value is obtained by an in-field expert of the studied system [122, 123]. Figure 9.3 shows two examples of fuzzy-sets for uncertain failure rate that is represented by a random variable x_k: triangular distribution $\mu^{tr}(x_k) = [a_k, \lambda_k, c_k]$, and Gaussian distribution $\mu^{Gaus}(x_k) = N(\lambda_k, \sigma_k]$, where λ_k is the similar referenced failure rate. The mathematical formulas of the two fuzzy sets are given in Equations (4) and (5) for triangular and Gaussian distribution respectively [124].

   The lower bound of the triangular fuzzy set is $a_k = (1 - Ef_k)\lambda_k$ and the upper bound is $a_k = (1 + Ef_k)\lambda_k$. The standard deviation of the Gaussian fuzzy set is $\sigma_k = Ef_k/3$.

$$\mu^{tr}(x_k) = [a_k, \lambda_k, c_k] = \begin{cases} \frac{x_k - a_k}{\lambda_k - a_k} & , a_k < x < \lambda \\ \frac{c_k - x_k}{c_k - \lambda_k} & , \lambda_k < x < c_k \\ 0 & o.w \end{cases} \tag{9.1}$$

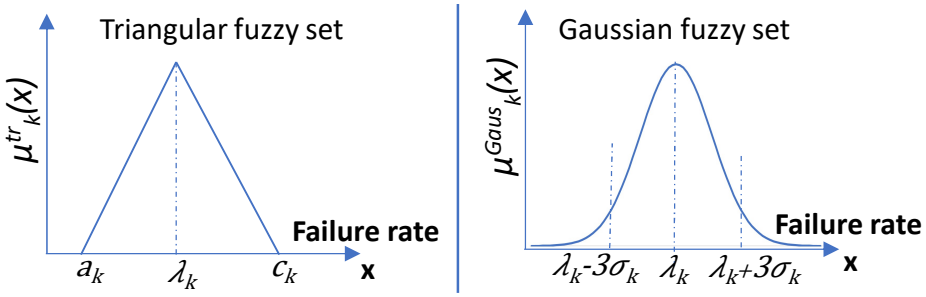


Figure 9.3 Triangular and Gaussian distribution Fuzzy sets.

$$\mu^{Gaus}(x_k) = N(\lambda_k, \sigma_k) = exp(-\frac{(x_k - \lambda_k)^2}{2\sigma_k^2}). \tag{9.2}$$

To present the data uncertainties for the example given in Figure 9.2, for 10% uncertainty and let $\lambda_1 = \lambda_2 = 0.0001$ Failure per hour (FPh), $\lambda_3 = 0.0000044$ FPh, and $\lambda_4 = 0.000002$ FPh. Four fuzzy sets are required to represent the four basic events. Each set is selected to be either a triangular or Gaussian fuzzy set. The triangular fuzzy set is simpler for analytical solving and has finite upper and lower bounds [55]. The triangular fuzzy sets are $x_1 \sim [a_1 = 9 \times 10^{-5}, \lambda_1 = 10^{-4}, c_1 = 1.1 \times 10^{-4}]$ for motor1, $x_2 \sim [a_2 = 9 \times 10^{-5}, \lambda_2 = 10^{-4}, c_2 = 1.1 \times 10^{-4}]$ for motor2,
$x_3 \sim [a_3 = 39.6 \times 10^{-7}, \lambda_3 = 44 \times 10^{-7}, c_3 = 48.4 \times 10^{-6}]$ for Battery failure,
and $x_4 \sim [a_4 = 1.8 \times 10^{-6}, \lambda_4 = 2 \times 10^{-6}, c_4 = 2.2 \times 10^{-6}]$ for switch failure.

3. Applying of the Fault-Tree logic rules
The fuzzy rules are the logical operations that satisfy the interconnection relations of the different levels of the system's Fault-Tree. The "AND" and "OR" gates are the most common logic gates used to build systems' fault-tree [122]. The output failure probability $F(t)_{AND}$ for k basic components/events that are connected in AND logic relation is given in Equation (9.3) [122].

$$\begin{aligned} F(t)_{AND} &= AND\{F_1(t), F_2(t), ..., F_k(t)\} \\ &= \Pi_{j=1}^{k} F_j(t) \end{aligned} \tag{9.3}$$

Meanwhile, the output failure probability $F(t)_{OR}$ at time t for k events/components that formulates OR logic relation is given in Equation (7) [110, 122].

$$\begin{aligned} F(t)_{OR} &= OR\{F_1(t), F_2(t), ..., F_k(t)\} \\ &= 1 - \Pi_{j=1}^{k}[1 - F_j(t)] \end{aligned} \tag{9.4}$$

The example given in Figure 9.2 is an electrical system and its components followed an exponential failure distribution [30, 109]. Due to the uncertainty factor, the failure probability of the k^{th} basic components is a function of the stochastic random variable x_k, and it is written as $x_k \sim [a_k, \lambda_k, c_k]$ for triangular sets. The failure Probability is given by $F_k(t) = 1 - e^{-x_k t}$. To apply the logic rules on the example in Figure 9.2 for 10000 working hours, the motors sub-system 1 failure is caused by motor1 AND motor2 failures. The failure probability of sub-system 1 is 1x3 vector, and it is calculated using Equation (9.3) as:
$F_1(t = 10000)_{AND} = \Pi_{k=1}^{2} F_k(t = 10000)$

$$\begin{aligned} &= \Pi_{k=1}^{2}[1 - e^{-[a_k, \lambda_k, c_k]t}] \\ &= [0.935, 0.865, 0.889] \end{aligned}$$

The sub-system 2 failure is the power failure, and its failure has two possible causes: Battery failure OR switch failure. The failure probability based OR logic relation is calculated using Equation (9.4) is given by: $F_2(t = 10000)_{OR} = 1 - \Pi_{k=3}^{4}[1 - F_k(t = 10000)]$

$$\begin{aligned} &= 1 - \Pi_{k=3}^{4} e^{-[a_k, \lambda_k, c_k]t} \\ &= [0.062, 0.056, 0.068] \end{aligned}$$

The No-Rotation system failure is caused by failures of subsystem 1 OR subsystem 2. For $N = 2$ subsystems and 10000 working hours, the system fuzzy set $F(t = 10000)_{set}$ of the system failure is 1x3 vector, and it is calculated using Equation (9.4) as:

$F(t = 10000)_{set} = 1 - \Pi_{N=1}^{2}[1 - F_N(t = 10000)]$

$$\begin{aligned} &= 1 - [1 - F_1(t = 10000)_{AND}][1 - F_2(t = 10000)_{OR}] \\ &= [0.061, 0.127, 0.1035] \end{aligned}$$

4. Defuzzification

   Defuzzification is the process of converting the system fuzzy set $F(t)_{set}$ into a deterministic crisp value $F(t)_{sys}$ that represents the system failure probability at a certain time t. In the reliability context, the centroid approach is the most common defuzzification technique [122]. The centroid is calculated as a center of the area under the system

fuzzy set $\mu_{sys}(x)$, as given in Equation (9.5) [30,109]. The failure rate of the integrated system λ_{sys} is represented by random variable x in the system fuzzy set.

$$F(t)_{sys} = \frac{\int_{-\infty}^{\infty} x_{sys}\mu_{sys}(x)\,dx}{\int_{-\infty}^{\infty} \mu_{sys}(x)\,dx} \tag{9.5}$$

For the example given in Figure 9.2 the defuzzification of the No-Rotation system failure is calculated using Equation (9.5) for triangular fuzzy set at 10000 working hours, as $\frac{1}{3}(a_{sys} + \lambda_{sys} + c_{sys})|_{t=10000} = 0.1$.
The system failure probability of the given example in Figure 9.2 at 10000 working hours and 10% uncertainty is 0.1. The system reliability $R(t)_{sys}$ is estimated at a certain time t and uncertainty factor Ef as $R(t)_{sys} = 1 - F(t)_{sys}$. The estimated system reliability of the example in Figure 9.2 at 10000 working hours and 10% uncertainty is 0.9.

9.3.2 Reliability Centered Maintenance (RCM) Activities.

The RCM activities are the actions that aim to avoid a failure mode and reduce the risk of multiple failures. These actions are mainly characterized as Condition-Based and Time-Based [30]. The condition-Based tasks are defined as Condition-Based Maintenance CBM strategy, and it is relayed on the real condition of the system/subsystems while analyzing its sensor data [30,55]. Time-Based tasks are the periodical planned replacement or restoration of a component, and it is referred as the Preventive Maintenance strategy [30, 55]. Time-directed tasks are suitable for systems/sub-systems that have low failure-variability and low failure propagation [30].The condition-direct activities are the most effective for critical systems, yet the system's operational and real-time data are a prerequisite [55]. Both RCM maintenance tasks of Preventive actions and Condition-Based actions are physical tasks. The human interaction plus repairing time are required to perform the maintenance task [125]. The developed scheme of this paper extends the RCM concept to include automatic actions that are possible to mitigate failure modes autonomously and increase the system's uptime. Figure 9.4 depicts the developed logic-Tree tool that identifies RCM action for each failure mode in Fault-Tree given in section 3.1. The selection of autonomous actions is specified by the dashed line rectangle in the logic Tree of Figure 9.4. The RCM worksheet is a table that contains mainly the RCM tasks to mitigate the failure mode or reduce its effects or consequences. The worksheet is not limited to RCM tasks, and it could contain more information to facilitate the implementation of RCM according to the studied system [109].
To improve the system's uptime autonomously, nine columns RCM worksheet sheet is proposed, and the columns are described as follows:

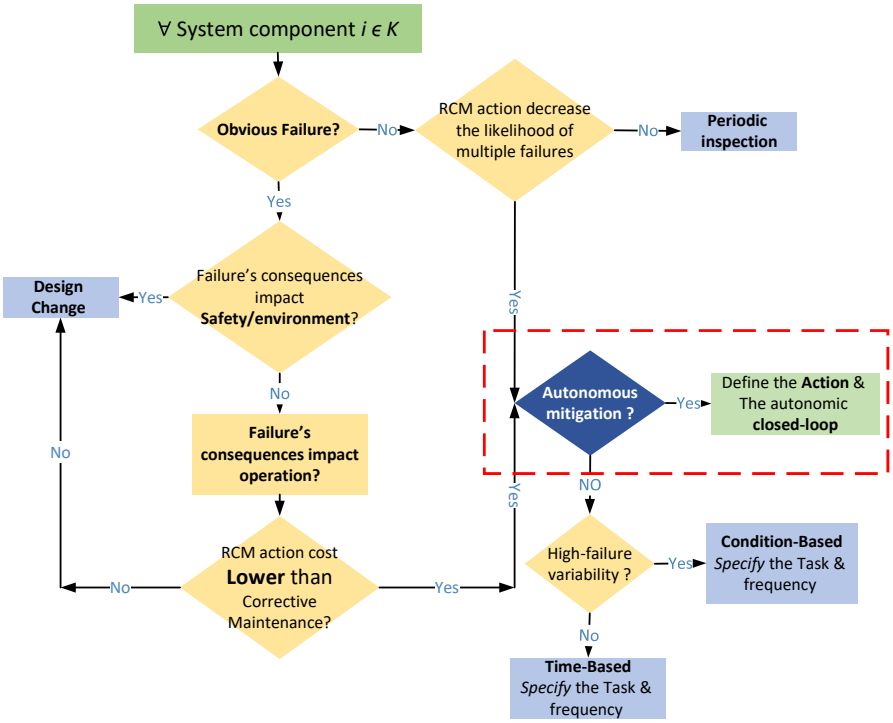


Figure 9.4 Logic-Tree of the Developed RCM Autonomous Action's selection.

1. **Index**: it indicates the index of the basic cause of failure, and it is the same index as given to the basic component in Fault-Tree.
2. **Component**: it indicates the basic component name.
3. **Failure mode**: it includes the failure mode that is linked to that component.
4. **Effects/Consequence**: it identifies the consequences and effects of that failure mode.
5. **Causes**: it defines the possible cause of having this failure mode.
6. **Detection**: it indicates the detection criteria of the failure mode if it is available.
7. **RCM Task**: it identifies the possible actions to avoid that failure mode or reduce its effect. The action type is selected for each basic component of the system according to the selection logic tree given in Figure 9.4.
8. **Frequency**: it indicates how often this RCM action is carried out.
9. **Autonomous Action**: this column is developed to define the possibility of mitigating the failure mode and increasing the system's uptime autonomously.These actions are

executed through the autonomic closed-loop between physical and digital environments.

For the example given in Figure 9.2, Table 9.2 demonstrate the possible RCM tasks according to the worksheet logic. The system example consists of two sub-systems: the motor failure sub-system and power failure sub-system. In this example, the cost of proactive maintenance is assumed to be lower than the cost of the corrective maintenance plus the failure consequence for each failure mode. Deployment of proactive maintenance tasks is assumed to decrease the likelihood of having multi-failure consequences. The root cause of the motor subsystem 1 failure is over-current, and the selected RCM action to mitigate this failure is a condition-directed action on a daily basis, according to the logic criteria in Figure 9.4. Meanwhile, the autonomous RCM action of discharging control is applicable to mitigate the battery's low-voltage in subsystem 2 while monitoring the battery discharging voltage. The switch failure could be identified for the design change task when the cost of the corrective maintenance and run to failure task are high, according to logic-tree criteria in Figure 9.4.

In the example given in Figure 9.2, Battery discharging control task is applicable to be executed through the DT autonomic closed loop to prevent the system disturbance.
To improve the uptime of the W.M.E-Bus autonomously, this paper addresses the RCM's actions that could be controlled through an autonomic closed loop. Section 4 verifies the proposed methodology on the on-road operation of the W.M.E-Bus under uncertainty of failure data.

9.4 Autonomous Scheme verification on Electric-Bus

The Wheel Motor Electric-Bus (W.M.E-Bus) drive systems consisted of Embedded Control Unite (ECU), and each wheel has a DC/AC 3-phase power inverter, and a 3-phase induction motor. Figure 9.5 shows the schematic diagram for the W.M.E-Bus drive system. The ECU controls the drive system and provides the control signals to the DC/AC inverters. It monitors the wheel motors' sensors and performs the driver's acceleration/deceleration

Table 9.2 Structure of the RCM Task worksheet

Index	Component	Failure mode	Consequence/ Symptoms	Example Possible Causes	Detection	RCM Task	Frequency	Autonomous action
1	Motor 1	Motor failure	No Rotation	Over current	NA	Current Monitoring	Daily	NA
2	Motor 2			Over current	NA	Current Monitoring	Daily	NA
3	Battery	Power failure		Low voltage	NA	Discharging Control	hourly	Yes
4	Switch			Wrong selection	NA	Change material	Once	NA

decision [110]. The 3-phase DC/AC inverter controls the induction motor speed and charges the battery in the motor's regenerative braking mode [120]. the developed scheme in this paper increases the reliability and uptime of the W.M.E-Bus's drive system in Figure 9.5. This system is categorized into six main subsystems, and each sub-system is dismantled to its basic components using the Fault-tree analysis approach in section 3.1. A detailed description of each basic component for the W.M.E-Bus drive system is given in [110]. Table 9.3 contains the W.M.E-Bus Subsystems, their indices, uncertain failure rate data in Failure per 10^6 hours (FPMh) for each subsystem.

The given data in Table 9.3 are gathered based on the basic components in the schematic of the W.M.E-Bus drive system in Figure 9.5, and the similar components in the literature [1, 2, 4, 110, 121].

9.4.1 Electric-Bus Drive System Health Monitoring

The Fuzzy-FTA (FFTA) approach is applied to analyze the system fault tree with the uncertainty of component failure rate. Figure 9.6 is the top three levels of the W.M.E-Bus drive system Fault-tree. It includes the subsystems that are given in Table 9.3. The lower levels of the basic components of each subsystem are presented by the blue triangle in Figure 9.6, and it is fully descried with its indices in Figure 9.7.

The uncertainty factor of the failure rate in the power electronics converters is between 10% and 24% [110, 119]. In the case of unavailability of the operational data, the system failure uncertainty is an expert-based factor [122, 123, 126]. Based on an expert recommendation from an e-mobility maintenance center, the uncertainty factor for the W.M.E-Bus is 17 %. This percentage is calculated as the difference between the designed allowable millage and the real failure miles for the W.M.E-Bus sub-systems. In this paper, the fault-tree in Figure 9.6 is analyzed using FFTA with Gaussian distribution fuzzy sets and 17% uncertainty.

9.4.2 Electric-Bus Drive System Autonomous Actions

The autonomous actions list is defined using the same concept of the RCM actions logic-tree in section 3.2. In this paper, the actions list addresses the actions that could be executed automatically through an autonomic closed loop. The autonomous actions worksheet for the W.M.E-Bus Drive system is built on the system Fault-tree given in Figure 9.6. The recent IIOT technology and Digital Twin (DT)'s connectivity to the E-Bus enable these autonomous actions to be carried out.

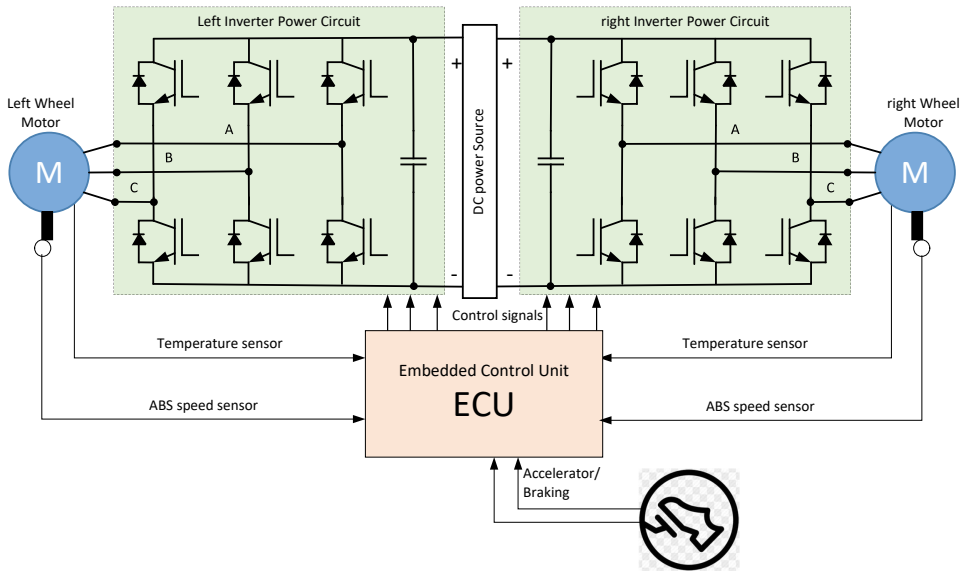


Figure 9.5 W.M.E-Bus Drive System Schematic.

In this section, for each sub-system, the extracted autonomous actions from the worksheet are discussed, and the full actions-list is given in Appendix C.

1. Embedded control Unite (ECU) Subsystem Failure.
   It is a DSP control unit, and its potential failure causes are embedded software bugs or the protection circuit component failure [110]. The autonomous action tackles the software issues that are indexed by C.1 in the system's Fault-Tree Figure 9.6. The software version of the W.M.E-Bus's ECU is automatically checked on daily bases. Once there is an update of the manufacturer's software database, DT updates the W.M.E-Buss' ECU by Over-The-Air (OTA) software update [127].

2. AC/DC 3-phase Inverter Subsystem Failure
   The power inverter sub-system failure is defined as a critical operational concern. This failure causes a malfunction of the electrical motor and inverters sub-systems failure. To increase the inverter's uptime autonomously, the Switching frequency check is the autonomous RCM action to mitigate the abnormal noise failure mode I.3 in Figure 9.7. Switching frequency is one of the causes that create an unacceptable vibration noise of the inverter, especially in the system integration of a different supplier [128, 129]. At each ECU software update, the compatibility of the inverter's frequency is checked autonomously.

Table 9.3 W.M.E-Bus Drive system and subsystems Failure rate

No	Driving System Components	symbol	(FPMh)
1	ECU failure	C	3.44E-06
Left Wheel Subsystems			
2	Inverter failure	I1	1.54E-06
3	Sensor's failure	S1	2.83E-07
4	Bearing failure at 11000 rpm	B1	1.74E-06
5	Mechanical-link failure	G1	6.11E-08
6	Motor windings failure	W1	3.00E-07
Right Wheel Subsystems			
7	Inverter failure	I2	1.54E-06
8	Sensor's failure	S2	2.83E-07
9	Bearing failure at 11000 rpm	B2	1.74E-06
10	Mechanical-link failure	G2	6.11E-08
11	Motor windings failure	W2	3.00E-07

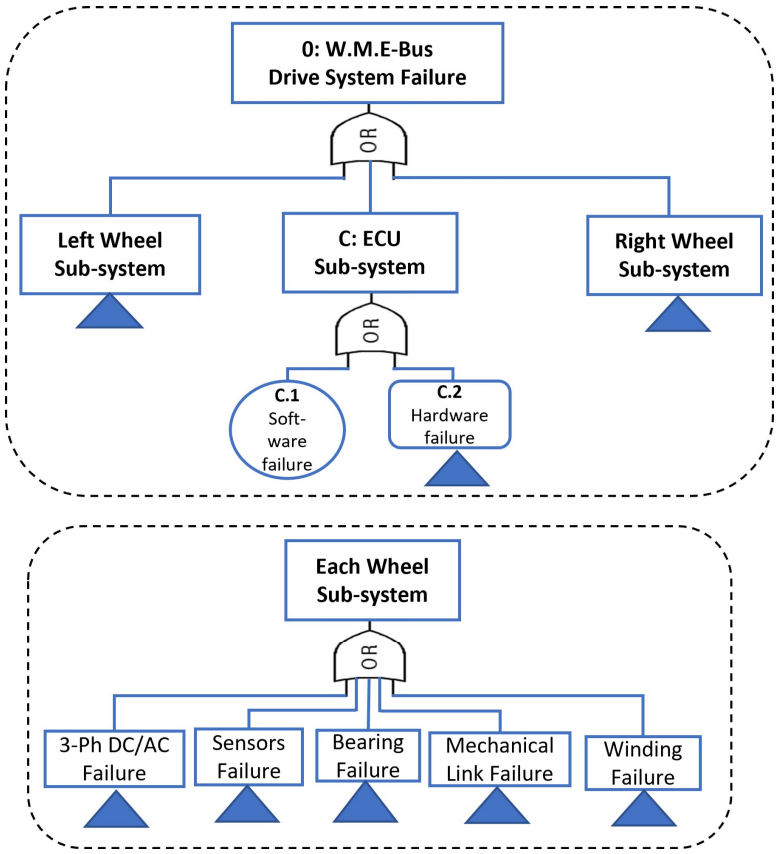


Figure 9.6 Top Three levels in the Fault-Tree of W.M.E-Bus Drive System.

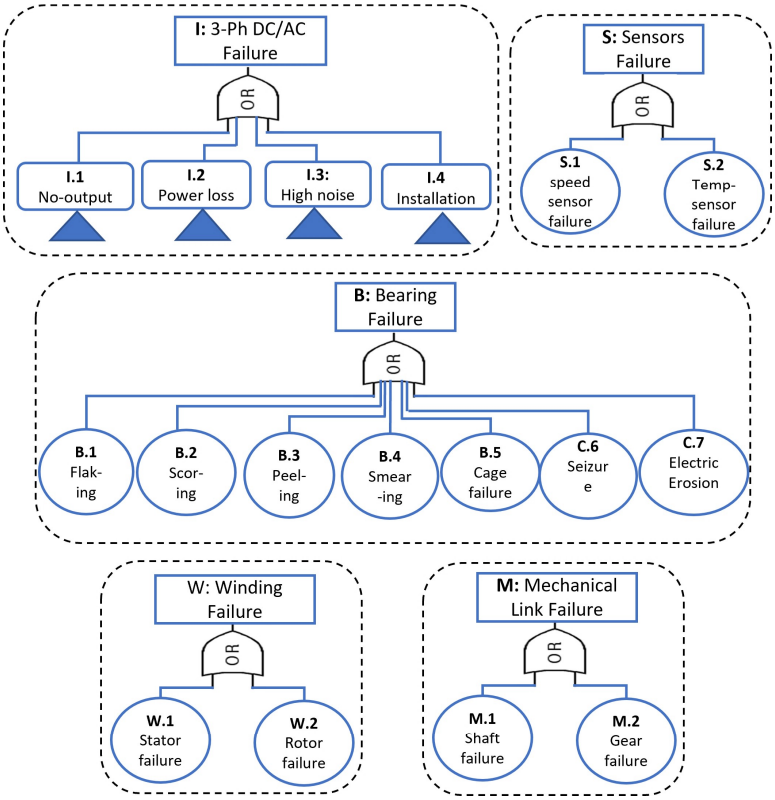


Figure 9.7 Fault-Tree's Basic Level of the W.M.E-Bus Drive system.

3. Three-phase Induction motor's Subsystem Failure

   Motor failure could be caused by the winding, the bearing, or the mechanical link sub-systems [110]. The worksheet in Appendix C includes the detailed RCM tasks for each subsystem's failure mode. From the perspective of autonomous improvement, the wheel motor's degradation is related to the internal mechanical and electrical power losses of the motor that caused internal thermal stress [117]. Reducing the motor's power losses P_{losses} (Kw) at the operation leads to reducing the degradation rate of the motor sub-system [117]. The autonomic loop is objected to decrease the wheel motor power losses at each operating cycle to increase the system uptime. Equation (9.6) is the losses over an operating cycle time T_{cycle}, and η is the motor efficiency according to its manufacture. The wheel motor output P_{out} (kw) in Equation (9.7) is related to the load torque T_{load} (Nm) and motor speed $n(t)$ (RPM).

$$P_{losses} = \int_0^{T_{cycle}} (\frac{1}{\eta} - 1) P_{out}\, dt \tag{9.6}$$

$$P_{out} = \frac{2\pi\ n(t)\ T_{load}}{60} \tag{9.7}$$

With fully loaded W.M.E-Bus conditions, the bearing is the most affected subsystem with the changing of the wheel motor speed [1,130,131]. Next section 4.3 links the real operating cycles of a W.M.E-Bus to the system analysis to improve its uptime. The full DT algorithm of the proposed scheme implementation is also included.

9.4.3 Electric-Bus Driving cycle

The driving cycle is a chart that plot the Electric-Bus speed versus running time, and it shows the acceleration and deceleration behaviors of the E-Bus on a certain route. The FTP-75 city driving, HWFET highway driving, and SFTP US06 aggressive driving are the standard driving cycles for testing light duties-vehicle in North America [132]. For heavy-duties, WVU West Virginia University and EPA Urban schedule are the recommended standard driving cycles [132]. In this paper, the developed approach is implemented on W.M.E-Bus in operation, and a practical on-road driving cycle is used to analyze the drive systems and their subsystems. Figure 9.8 shows an example of real on-road W.M.E-Bus driving cycle and W.M.E-Bus speed $V(t)$ is in mile/h. The electrical motor speed $n(t)$ (RPM) is abstracted by Equation (9.8), where the $V(t)$ is the electric-bus (wheel end) speed in mile/h. The gear ratio is G for the transmission gears from the motor to the wheel, and the wheel's tire radius is r (m). The motor speed $n(t)$ (rpm) over the practical driving cycle is given in Figure 9.9.

$$n(t) = \frac{60V(t)}{2\pi rG} \tag{9.8}$$

The average motor speed over the driving cycle in Figure 9.9 is 2353.95 RPM. The operating speeds of the inverter-driven motors are influenced its failure analysis and consequently the reliability analysis of E-Bus system. The bearing of the electrical motor is the main affected motor component with the operating speed [1, 130, 131]. The bearing reference failure rate λ_B (Failures/Mh) is given by Equation (9.9) [1,110]. C_{env} is a multiplication factor that represents the bearing environmental condition (ex. Environmental temperature and lubricant) this factor is calculated based on charts and table in [1]. In the case study of this paper, the W.M.E-Bus works in the North America region and the environment multiplication factor equals 1.716.

$$\lambda_B = \lambda_{BE,B} C_{env} \tag{9.9}$$

$\lambda_{BE,B}$ is the basic bearing failure rate (Failures/Mh) given by Equation (9.10), Where L_{10} is the standard bearing life in (milion rev) [1]. Equation (9.11) indicates the operational failure

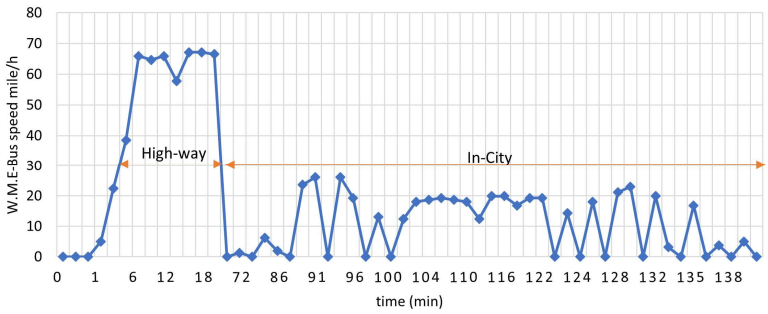


Figure 9.8 On-Road Electric-Bus Drive Cycle with Bus speed in (mile/h).

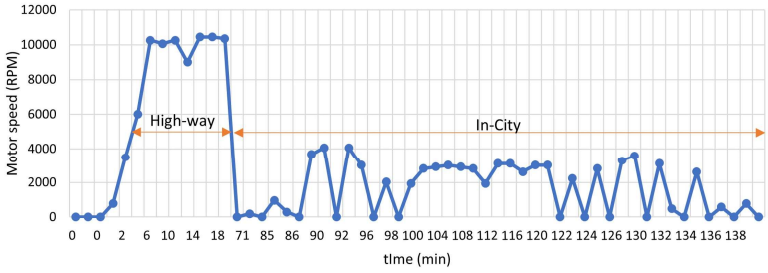


Figure 9.9 Wheel Motor Speed (rpm) Over the Driving Cycle in Figure 9.8.

rate for the motor bearing $\lambda_{B)op}$ (Failures/Mh.rpm). The developed formula in Equation (9.11) establish the relation between W.M.E-Bus's bearing failure rate and the operational motor speed (rpm).

$$\lambda_{BE,B} = 2 \times 10^5 [\frac{60\ n(t)}{L_{10}}] \tag{9.10}$$

$$\lambda_{B)op} = \frac{120 \times 10^5\ n(t)\ C_{env}}{L_{10}} \tag{9.11}$$

As the bearing failure rate has a direct relation to the wheel motor operating speed, the operational reliability analysis of W.M.E-Bus depends on the bus driving cycle. The Proposed algorithm in Figure 9.10 checks the daily W.M.E-Bus reliability and manipulates the driving cycle for the next day.

In the form of the bus's running miles, the operating time t (h) is converted to distances d (mile) using the average of the nominal on-road driving cycle given in Figure 9.9. The W.M.E-Bus drive system reliability at d (mile) is $R(d)$ and the failure probability is $F(d)$. In the Digital Twin (DT), the failure and reliability analysis of the W.M.E-Bus system are executed using the FFTA method in section 3. The proposed DT algorithm reads the W.M.E-Bus driving cycle, and then estimate the system reliability.

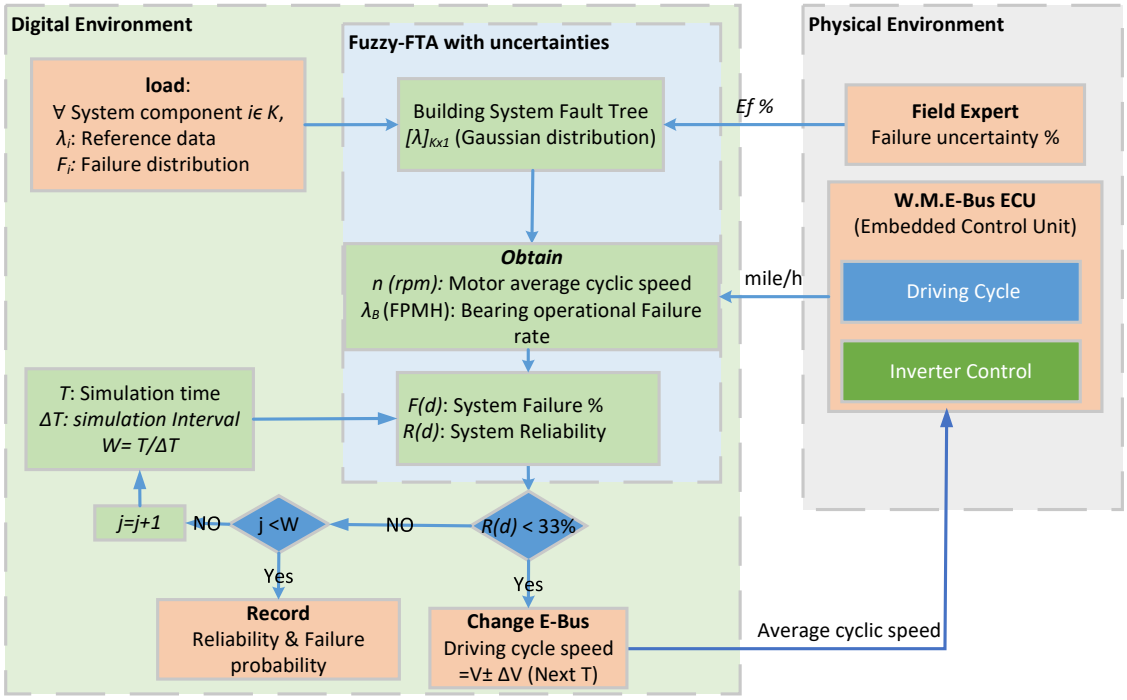


Figure 9.10 E-Bus Autonomous Uptime Improvement Algorithm.

In case of having lower reliability for example lower than 33%, the DT connects to the Bus's ECU to manipulate the average cyclic driving speed and improve the system's reliability and increase its uptime on the next day of operation. The critical reliability level is adjusted according to the fleet management strategies. In the studied case given by Figure 9.9, the W.M.E-Bus is inspected every 600 K mile which is equivalent to 33% system reliability and 67% failure probability. The simulation cycle time ΔT is selected to be one day (24-hour service). The sub-systems failure uncertainty factor is obtained based on the maintenance workshop of the W.M.E-Bus driving system and it could be updated with the ongoing physical replacement tasks. The achieved uptime-improvements with the proposed algorithm are indicated in the results section.

9.5 Result and discussion

The DT algorithm in Figure 9.11 is the implementation of the proposed scheme of autonomous uptime-improvement for the Wheel Motor Electric Bus (W.M.E-Bus). This algorithm is verified with Gaussian fuzzy sets to present the sub-systems failure data uncertainty and an expert-based uncertainty factor of 17 %. For the studied case given in Figure 9.9, the 17 % uncertainty factor is obtained based on the difference between the real failure miles and

designed one for the W.M.E-Bus drive system,

For W.M.E-Bus drive system analysis with 17% failure data uncertainties, Table 9.4 compares the performance of the proposed Gaussian defuzzification method versus Mount Carlo (MC) and Latin Hypercubic (LH) methods that sampled from a Gaussian distribution [124]. This comparison is carried out for the W.M.E-Bus drive system that runs on the conditions of the on-road driving cycle in Figure 9.9, and the results are reported at a running mileage of 400,000 miles. The proposed method of defuzzification of Gaussian fuzzy sets performs as the LH sampling method with 100 samples and 17 % uncertainty factor.

Figure 9.11 shows the W.M.E-Bus system reliability in the blue line and failure probability in the orange line. At 600,000 running miles, the reliability of W.M.E-Bus drive system degrades to 33% with the nominal on-road driving cycle given in Figure 9.9.

To improve the system reliability during the operation, the proposed DT algorithm in Fig-

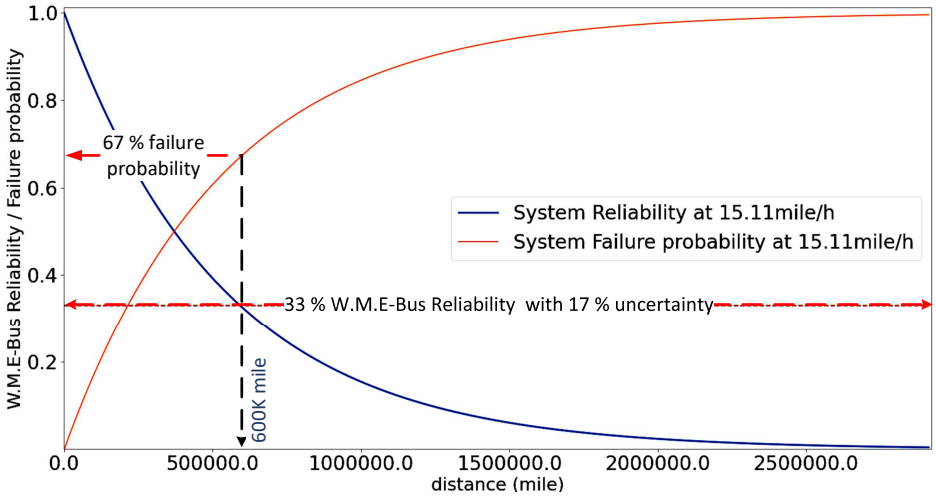


Figure 9.11 FFTA W.M.E-Bus system analysis at the average cyclic speed of 15.11 mile/h.

ure 9.10 will change the controllable variable of the W.M.E-Bus; speed over the driving cycle for the next operating day. Figure 9.12 shows the W.M.E-Bus drive system reliability at

Table 9.4 WME-Bus Analysis at 15 mile/h nominal average cyclic speed and 400K mile for different uncertainty analysis techniques

Index	**Triangle Defuzzification**	**Gaussian Defuzzification**	**MC sampling [22] 100 samples**	**LHS sampling [22] 100 samples**
Failure Probability	0.52662931	0.526646	0.53319864	0.52659089
System Reliability	0.48448179	0.484464	0.47791246	0.48451

different operating driving cycle speeds. At the same running distance of 600,000 miles, the reliability of the W.M.E-Bus drive system increases as the developed algorithm manipulates the bus's driving cycle and decreases the operating speeds.

In this case study, the allowable W.M.E-Bus mileage distance before performing the pre-

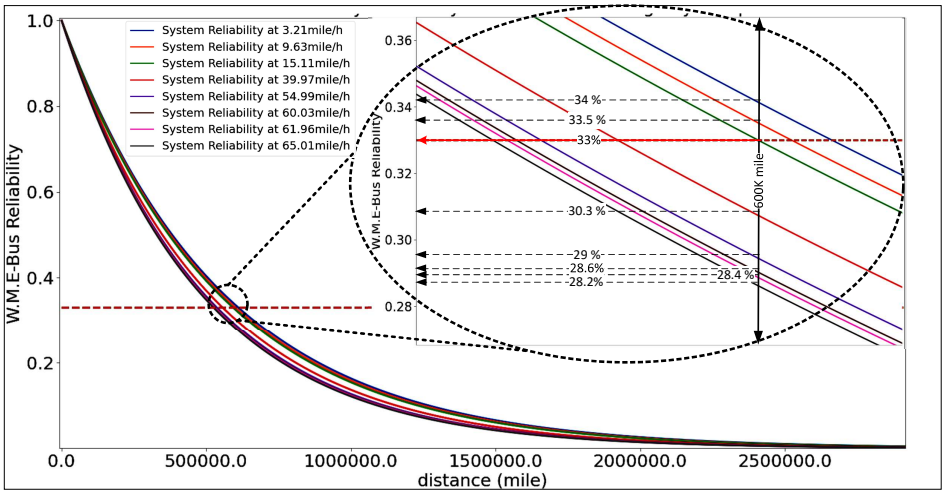


Figure 9.12 W.M.E-Bus Reliability analysis at different operating E-Bus speeds.

ventive maintenance is 600,000 miles. Figure 9.13 demonstrates the impacts of the proposed scheme on the allowable W.M.E-Bus mileage distances at different operating average cyclic speeds. At 33% system reliability, the allowable W.M.E-Bus's mileage increases as the pro-

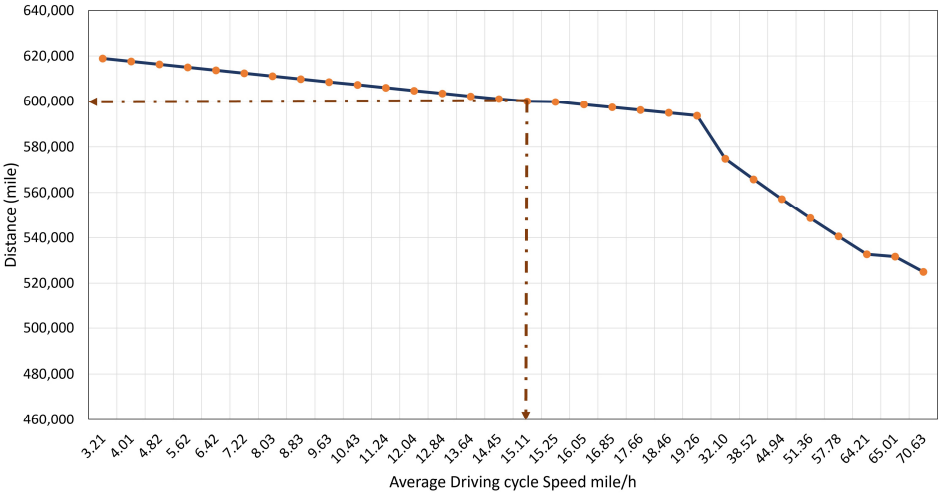


Figure 9.13 W.M.E-Bus Allowable Distance (mile) Versus Average Driving Cycle speed (mile/h) at Same Level of Reliability.

posed DT algorithm interacts with the ECU of the E-Bus and slowdown the bus's driving cycle. For example, changing operating speeds from 15.11 mile/h to 9.63 mile/h increases the allowable bus's mileage by 8603 miles over the 600,000 miles that is adjusted as the maximum allowable millage for the studied case.

9.6 Conclusion

This paper proposed a scheme for the implementation of autonomous uptime-improvement systems in a Digital Twin (DT) environment while the operational in-field data are unavailable. The Basics of the Reliability Centered Maintenance (RCM) is extended to develop a DT scheme that is capable to improve the system uptime autonomously. The Fuzzy Fault Tree analysis is developed to analyze the system reliability with uncertainty of data. The proposed scheme is verified on Wheel Motor Electric Bus (W.M.E-Bus) drive system. To overcome the unavailability of the operational data, the W.M.E-Bus drive system is dismantled to subsystems and basic components to analysis the system reliability. The developed DT scheme links the uptime analysis with the real operating condition of the W.M.E-Bus while using on-road driving cycle. The uptime-improvement of the W.M.E-Bus is measured and compared to the existing inspection criteria, and the proposed scheme adds 8603 miles to the maximum allowable mileage distance when derating the average operating speed by 36%. In Future work, implementation of the decision making, Self-healing mechanism need to be addressed. The developed scheme will extend to include operation cost, battery degradation, and the DC power system.

CHAPTER 10 GENERAL DISCUSSION

The fourth industrial revolution relays on digitized and fully connected machines. This digital transformation and connectivity enable the observability and controllability of the machines. According to these hypotheses, this thesis develops different intelligent approaches in a Digital Twin (DT) environment to improve the machines' uptime.

This thesis developed different approaches to implementing the self-healing and autonomously maintained machine. Different industrial systems were studied with different types of data, including time-series and time-independent numerical data. Several Machine Learning (ML) algorithms were developed to analyze the system sensor data and monitor its degradation state. These algorithms modeled the health monitoring module that detects, diagnosis, and predict system failures. Another module is developed to act automatically online to increase the uptime while manipulating the system settings. The integration of these two modules formulates the closed-loop of the autonomous system.

The performance degradation of a system can be stopped if it is detected and suitable required actions are applied. In the degradation monitoring process, the system's faults/failures are detected and diagnosed to trigger the self-healing mechanism. Integration of Long-Short-Term-Memory (LSTM) and Regression Adjustment for Multivariate (RAM) detects the positional anomalies of a robotic arm with correlated and unsupervised data variables. Meanwhile, in the Pre-failure approaches, the monitoring module should include the Remaining Useful Life (RUL) prediction or the Time to Failure (T2F) estimation of the system.

The logical Analysis of Data (LAD) pattern recognition technique enables the post-failure self-healing mechanism. The online health monitoring module detects the instant of faults, and it is modeled by the generated patterns when ingesting the uncontrollable variable data to the LAD algorithm. Then, the self-healing module provides corrective settings to clear the detected fault. The recovery patterns are generated when ingesting the controllable variable data in the machine's normal operation to the LAD algorithm. For fast recovery, Distance approaches were the criteria of the self-healing module to select the nearest recovery pattern to the machine's settings at the instant of the fault.

Reinforcement Learning (RL) is a model-free approach and it has low online computation. The combination of RL and LAD develops an online Pre-failure autonomous slowdown for the system's degradation. The online health monitoring module is modeled by the generated LAD patterns when ingesting the run-to-failure time series data. This module is able to monitor and detect the Pre-Failure interval of the system while tracking the P-F curve.

Once, the Potential failure point is detected, the Pre-Failure RL module is activated and generates continuous autonomous actions according to the trained optimal policy. These actions increase the Time to Failure (T2F) while keeping the system performance at an acceptable level.

With the lack of system operational data, Fuzzy Fault-Tree Analysis (FFTA) algorithm is adopted to analyze the Electric Bus (E-Bus) with the uncertainty of system failure data. This approach estimates the system failure system and defines its critical components. The concept of Reliability Centered Maintenance is adopted and extended to enable the autonomous uptime improvement for the E-Bus with the unavailability of data or data uncertainty. Integration of FFTA and RCM's actions in a DT environment develops an autonomous scheme that increases the E-Bus uptime according to the on-road real driving conditions.

CHAPTER 11 CONCLUSION AND RECOMMENDATIONS

11.1 Summary of Works

This thesis has presented novel approaches to implementing autonomously maintained machines. The main objective of the research was uptime improvement. This thesis addressed the Time to Failure (T2F) extension and self-healing mechanisms to improve the performance of machines. The approaches developed in this thesis verified the two limbs of the autonomous systems concept in maintenance: system health monitoring, detection, and diagnosis, and (2) autonomous improvement actions. The health monitoring and diagnostic modules are developed for different systems/assets to provide a system degradation state to the autonomous action module. According to this state, the proper autonomous action is performed to increase the system uptime performance. In this thesis, supervised and non-supervised machine learning algorithms were developed to build the health monitoring and autonomous recovery modules. The impact of the autonomous module on system performance is validated in different industrial systems.

The industrial data of the studied systems are an essential part of learning and developing an autonomous system. The data variables have to cover the non-controllable variables of a system to be fully monitored, and the setting variables of the system to fully controlled. The data type defines the main objectives of the health monitoring and autonomous recovery modules. For example, time-independent data of faulty and non-faulty observations are useful in building a post-failure self-healing mechanism to recover the machine at the instant of a fault. This mechanism is implemented on a CNC milling machine and it was capable of recovering the machine from a non-conformation (faulty) state to a conformation (non-faulty) state. The health monitoring module detects undesired performance (fault/anomalies) and enables the self-healing mechanism to recover the machine. cbmLAD Pattern recognition machine learning algorithm was a successful method to analyze this supervised data and to develop the novel self-healing mechanism.

In another application, Time series run-to-failure data was used to develop a pre-failure tool performance improvement approach. a tool degeneration state is monitored online and the potential failure level is estimated. A deep RL agent is developed to slowdown the tool degradation rate while manipulating the machine settings according to the optimal trained policy. The implementation of this approach in the particular case study increased Time to Failure (T2F) by 40% on average, compared to the classical approach.

If the assets data is unavailable, the autonomous uptime improvement approach was developed on the basis of reliability Centered Maintenance (RCM) and failure analysis of uncertain data. A novel scheme was developed to increase the uptime of the Electric-Bus (E-Bus) in Digital Twin. The drive system of the E-Bus was analyzed to estimate its reliability, and according to the real driving cycle, the autonomous action is performed to extend the allowable millage. Deployment of the developed structure could increase the maximum allowable E-Bus millage by 8603 miles on average.

11.2 Future Work

In future research, the validation of physical hardware systems needs to be addressed. More physical experiments needed to be conducted to verify the IIOT send/receive responses and the implementation of real-world autonomous action . The self-healing mechanism developed addressed the anomalies of the systems in the post-failure mode. To have a robust self-healing mechanism, more analysis and experiments are required. The mechanism needs to be extended to include the failure of sensors and actuators, for example having diagnostic and availability prediction software for sensors and actuators.

A generalized autonomous healing approach for pre-failure and post-failure is needed, to extend the feasibility of the health monitoring and the autonomous action modules while using transfer learning. In this machine learning technique, the developed models could be used to perform another application. Transfer learning enables the machine learning model to train on one application and transfer the training knowledge to any other similar application.

Continuous online learning stimulates the explorations of new failures or new actions. The incremental-learning AI modules enable continuous updating of the health monitoring and autonomous action models. The autonomous action module could learn more action space, and the health monitoring will train on more faults. Consequently, The autonomous modules will be adaptable to any changes in the environment of the machine.

REFERENCES

N. S. W. Center, *Handbook of reliability prediction procedures for mechanical equipment*. Carderock Division, Naval Surface Warfare Center, 1992.

X. Shu, Y. Guo, W. Yang, K. Wei, Y. Zhu, and H. Zou, "A detailed reliability study of the motor system in pure electric vans by the approach of fault tree analysis," *IEEE Access*, vol. 8, pp. 5295–5307, 2019.

N. Choudhury, "Reliability prediction of electronic products combining models, lab testing and field data analysis," 2016.

J.-J. Baek, "A study on reliability evaluation of embedded software in vehicle," *Transactions of the Korean Society of Automotive Engineers*, vol. 19, no. 4, pp. 1–7, 2011.

J. Lee, M. Ghaffari, and S. Elmeligy, "Self-maintenance and engineering immune systems: Towards smarter machines and manufacturing systems," *Annual Reviews in Control*, vol. 35, no. 1, pp. 111–122, 2011.

A. A. Amin and K. M. Hasan, "A review of fault tolerant control systems: advancements and applications," *Measurement*, vol. 143, pp. 58–68, 2019.

A. K. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical Systems and Signal Processing*, vol. 20, no. 7, pp. 1483 – 1510, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0888327005001512

M. D. Holloway and E. Holloway, *Dictionary of Industrial Terminology*. John Wiley & Sons, 2013.

P. Mhaskar, J. Liu, and P. D. Christofides, *Fault-tolerant process control: methods and applications*. Springer Science & Business Media, 2012.

J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," *Manufacturing Letters*, vol. 3, pp. 18–23, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S221384631400025X

W.-J. Zhang and Y. Lin, "On the principle of design of resilient systems–application to enterprise information systems," *Enterprise Information Systems*, vol. 4, no. 2, pp. 99–110, 2010.

E. Glaessgen and D. Stargel, "The digital twin paradigm for future nasa and us air force vehicles," in *53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA*, 2012, p. 1818.

Q. Qi, F. Tao, T. Hu, N. Anwer, A. Liu, Y. Wei, L. Wang, and A. Nee, "Enabling technologies and tools for digital twin," *Journal of Manufacturing Systems*, vol. 58, pp. 3–21, 2021.

A. AboElHassan, A. Sakr, and S. Yacout, "A framework for digital twin deployment in production systems," in *Advances in Automotive Production Technology – Theory and Application*, P. Weißgraeber, F. Heieck, and C. Ackermann, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2021, pp. 145–152.

G. W. Vogl, B. A. Weiss, and M. Helu, "A review of diagnostic and prognostic capabilities and best practices for manufacturing," *Journal of Intelligent Manufacturing*, vol. 30, no. 1, pp. 79–95, 2019.

H. A. Taha, S. Yacout, and Y. Shaban, "Autonomous self-healing mechanism for a cnc milling machine based on pattern recognition," *Journal of Intelligent Manufacturing*, pp. 1–21, 2022.

S. Khan, M. Farnsworth, R. McWilliam, and J. Erkoyuncu, "On the requirements of digital twin-driven autonomous maintenance," *Annual Reviews in Control*, vol. 50, pp. 13–28, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1367578820300560

H. A. Taha, S. Yacout, and L. Birglen, "Detection and monitoring for anomalies and degradation of a robotic arm using machine learning," in *Advances in Automotive Production Technology–Theory and Application*. Springer, 2021, pp. 230–237.

Y. Zhang and J. Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems," *Annual reviews in control*, vol. 32, no. 2, pp. 229–252, 2008.

Y. Shaban, M. Meshreki, S. Yacout, M. Balazinski, and H. Attia, "Process control based on pattern recognition for routing carbon fiber reinforced polymer," *Journal of Intelligent Manufacturing*, vol. 28, no. 1, pp. 165–179, 2017.

A. Elsheikh, S. Yacout, M.-S. Ouali, and Y. Shaban, "Failure time prediction using adaptive logical analysis of survival curves and multiple machining signals," *Journal of Intelligent Manufacturing*, vol. 31, no. 2, pp. 403–415, 2020.

H.-S. Park and N.-H. Tran, "Development of a smart machining system using self-optimizing control," *The International Journal of Advanced Manufacturing Technology*, vol. 74, no. 9, pp. 1365–1380, 2014.

S. Yacout, "Industrial value chain research and applications for industry 4.0," in *4th North America conference on Industrial Engineering and Operations Management, Toronto, Canada*, 2019.

J. Lee, H. D. Ardakani, S. Yang, and B. Bagheri, "Industrial big data analytics and cyber-physical systems for future maintenance service innovation," *Procedia CIRP*, vol. 38, pp. 3–7, 2015, proceedings of the 4th International Conference on Through-life Engineering Services. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827115008744

B. Bagheri, S. Yang, H.-A. Kao, and J. Lee, "Cyber-physical systems architecture for self-aware machines in industry 4.0 environment," *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 1622–1627, 2015.

A. Elsheikh, S. Yacout, and M.-S. Ouali, "Bidirectional handshaking lstm for remaining useful life prediction," *Neurocomputing*, vol. 323, pp. 148–156, 2019.

M. Yuan, Y. Wu, and L. Lin, "Fault diagnosis and remaining useful life estimation of aero engine using lstm neural network," in *2016 IEEE International Conference on Aircraft Utility Systems (AUS)*, Oct 2016, pp. 135–140.

Y. Shaban, S. Yacout, M. Balazinski, and K. Jemielniak, "Cutting tool wear detection using multiclass logical analysis of data," *Machining Science and Technology*, vol. 21, no. 4, pp. 526–541, 2017.

S. R. Barde, S. Yacout, and H. Shin, "Optimal preventive maintenance policy based on reinforcement learning of a fleet of military trucks," *Journal of Intelligent Manufacturing*, vol. 30, no. 1, pp. 147–161, 2019.

J. Moubray, *Reliability-centered maintenance*. Industrial Press Inc., 2001.

M. Lejeune, V. Lozin, I. Lozina, A. Ragab, and S. Yacout, "Recent advances in the theory and practice of logical analysis of data," *European Journal of Operational Research*, vol. 275, no. 1, pp. 1–15, 2019.

A. Bennane and S. Yacout, "Lad-cbm; new data processing tool for diagnosis and prognosis in condition-based maintenance," *Journal of Intelligent Manufacturing*, vol. 23, no. 2, pp. 265–275, 2012.

Y. Ma, W. Zhu, M. G. Benton, and J. Romagnoli, "Continuous control of a polymerization system with deep reinforcement learning," *Journal of Process Control*, vol. 75, pp. 40–47, 2019.

T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

S. S. P. Kumar, A. Tulsyan, B. Gopaluni, and P. Loewen, "A deep learning architecture for predictive control," *IFAC-PapersOnLine*, vol. 51, no. 18, pp. 512–517, 2018.

A. Traue, G. Book, W. Kirchgässner, and O. Wallscheid, "Toward a reinforcement learning environment toolbox for intelligent electric motor control," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

T. De Bruin, J. Kober, K. Tuyls, and R. Babuska, "Experience selection in deep reinforcement learning for control," *Journal of Machine Learning Research*, vol. 19, 2018.

Y. Zhang, W. Zhu, and A. Rosendo, "Qr code-based self-calibration for a fault-tolerant industrial robot arm," *IEEE Access*, vol. 7, pp. 73 349–73 356, 2019.

Accuracy Degradation Analysis for Industrial Robot Systems, ser. International Manufacturing Science and Engineering Conference, vol. Volume 3: Manufacturing Equipment and Systems, 06 2017. [Online]. Available: https://doi.org/10.1115/MSEC2017-2782

M. Goel, A. A. Maciejewski, and V. Balakrishnan, "Analyzing unidentified locked-joint failures in kinematically redundant manipulators," *Journal of Robotic Systems*, vol. 22, no. 1, pp. 15–29. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20046

J. Qin, F. Léonard, and G. Abba, "Real-time trajectory compensation in robotic friction stir welding using state estimators," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 6, pp. 2207–2214, 2016.

G. Qiao, C. Schlenoff, and B. A. Weiss, "Quick positional health assessment for industrial robot prognostics and health management (phm)," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1815–1820.

F. Piltan, C.-H. Kim, and J.-M. Kim, "Advanced adaptive fault diagnosis and tolerant control for robot manipulators," *Energies*, vol. 12, no. 7, 2019. [Online]. Available: https://www.mdpi.com/1996-1073/12/7/1281

M. A. Costa, B. Wullt, M. Norrlöf, and S. Gunnarsson, "Failure detection in robotic arms using statistical modeling, machine learning and hybrid gradient boosting," *Measurement*, vol. 146, pp. 425–436, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263224119306050

Q. Yang, X. Li, Y. Wang, A. Ainapure, and J. Lee, "Fault diagnosis of ball screw in industrial robots using non-stationary motor current signals," *Procedia Manufacturing*, vol. 48, pp. 1102–1108, 2020, 48th SME North American Manufacturing Research Conference, NAMRC 48. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2351978920316036

M. R. Berthold and F. Höppner, "On clustering time series using euclidean distance and pearson correlation," *arXiv preprint arXiv:1601.02213*, 2016.

D. C. Montgomery, *Introduction to statistical quality control*. John Wiley & Sons, 2007.

C. A. Lowry and D. C. Montgomery, "A review of multivariate control charts," *IIE transactions*, vol. 27, no. 6, pp. 800–810, 1995.

L. Prechelt, "Automatic early stopping using cross validation: quantifying the criteria," *Neural Networks*, vol. 11, no. 4, pp. 761–767, 1998. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608098000100

W. Luo, T. Hu, C. Zhang, and Y. Wei, "Digital twin for cnc machine tool: modeling and using strategy," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 3, pp. 1129–1140, 2019.

J. Wang, L. Ye, R. X. Gao, C. Li, and L. Zhang, "Digital twin for rotating machinery fault diagnosis in smart manufacturing," *International Journal of Production Research*, vol. 57, no. 12, pp. 3920–3934, 2019.

G. C. Silva, E. E. Carvalho, and W. M. Caminhas, "An artificial immune systems approach to case-based reasoning applied to fault detection and diagnosis," *Expert Systems with Applications*, vol. 140, p. 112906, 2020.

T.-D. Hoang, Q.-V. Nguyen, V.-C. Nguyen, and N.-H. Tran, "Self-adjusting on-line cutting condition for high-speed milling process," *Journal of Mechanical Science and Technology*, vol. 34, no. 8, pp. 3335–3343, 2020.

M. H. Karray, F. Ameri, M. Hodkiewicz, and T. Louge, "Romain: Towards a bfo compliant reference ontology for industrial maintenance," *Applied Ontology*, vol. 14, no. 2, pp. 155–177, 2019.

H. A. Taha, A. H. Sakr, and S. Yacout, "Aircraft engine remaining useful life prediction framework for industry 4.0," in *4th North America conference on Industrial Engineering and Operations Management, Toronto, Canada*, 2019.

Y. Shaban, S. Yacout, and M. Balazinski, "Tool wear monitoring and alarm system based on pattern recognition with logical analysis of data," *Journal of manufacturing science and engineering*, vol. 137, no. 4, 2015.

A. T. Abbas, M. Abubakr, A. Elkaseer, M. M. E. Rayes, M. L. Mohammed, and H. Hegab, "Towards an adaptive design of quality, productivity and economic aspects when machining aisi 4340 steel with wiper inserts," *IEEE Access*, vol. 8, pp. 159 206–159 219, 2020.

——, "Towards an adaptive design of quality, productivity and economic aspects when machining aisi 4340 steel with wiper inserts," *IEEE Access*, vol. 8, pp. 159 206–159 219, 2020.

F. Cus, U. Zuperl, and V. Gecevska, "High speed end-milling optimisation using particle swarm intelligence," *Journal of Achievements in Materials and Manufacturing Engineering*, vol. 22, no. 2, pp. 75–78, 2007.

S. Huang, K. K. Tan, G. S. Hong, and Y. San Wong, "Cutting force control of milling machine," *Mechatronics*, vol. 17, no. 10, pp. 533–541, 2007.

U. Zuperl, F. Cus, and M. Reibenschuh, "Modeling and adaptive force control of milling by using artificial techniques," *Journal of Intelligent Manufacturing*, vol. 23, no. 5, pp. 1805–1815, 2012.

A. Sadek, M. Hassan, and M. Attia, "A new cyber-physical adaptive control system for drilling of hybrid stacks," *CIRP Annals*, vol. 69, no. 1, pp. 105–108, 2020.

Z. Wu, Y. Wu, T. Chai, and J. Sun, "Data-driven abnormal condition identification and self-healing control system for fused magnesium furnace," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 3, pp. 1703–1715, 2014.

H.-S. Park and N.-H. Tran, "Development of a smart machining system using self-optimizing control," *The International Journal of Advanced Manufacturing Technology*, vol. 74, no. 9-12, pp. 1365–1380, 2014.

S. Seguy, T. Insperger, L. Arnaud, G. Dessein, and G. Peigné, "On the stability of high-speed milling with spindle speed variation," *The International Journal of Advanced Manufacturing Technology*, vol. 48, no. 9-12, pp. 883–895, 2010.

L. C. Moreira, W. Li, X. Lu, and M. E. Fitzpatrick, "Supervision controller for real-time surface quality assurance in cnc machining using artificial intelligence," *Computers & Industrial Engineering*, vol. 127, pp. 158–168, 2019.

J. Yang, S. Lu, and L. Wang, "Fused magnesia manufacturing process: a survey," *Journal of Intelligent Manufacturing*, vol. 31, no. 2, pp. 327–350, 2020.

S. Kumar, T. W. Chow, and M. Pecht, "Approach to fault identification for electronic products using mahalanobis distance," *IEEE Transactions on Instrumentation and Measurement*.

C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the surprising behavior of distance metrics in high dimensional space," in *International conference on database theory*. Springer, 2001, pp. 420–434.

D. Jouan-Rimbaud and R. De Maesschalck, "The mahalanobis distance," *Chemom. Intellig. Lab. Syst*, vol. 50, pp. 1–18, 2000.

K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification." *Journal of machine learning research*, vol. 10, no. 2, 2009.

J. Li, S. Fong, R. K. Wong, and V. W. Chu, "Adaptive multi-objective swarm fusion for imbalanced data classification," *Information Fusion*, vol. 39, pp. 1–24, 2018.

J. Brownlee, *Deep learning with Python: develop deep learning models on Theano and TensorFlow using Keras*. Machine Learning Mastery, 2016.

M. Meshreki, A. Sadek, and M. Attia, "High speed routing of woven carbon fiber reinforced epoxy laminates," in *ASME International Mechanical Engineering Congress and Exposition*, vol. 45196. American Society of Mechanical Engineers, 2012, pp. 2061–2066.

S. Rawat and H. Attia, "Characterization of the dry high speed drilling process of woven composites using machinability maps approach," *CIRP annals*, vol. 58, no. 1, pp. 105–108, 2009.

F. Koenig, P. A. Found, and M. Kumar, "Innovative airport 4.0 condition-based maintenance system for baggage handling dcv systems," *International Journal of Productivity and Performance Management*, vol. 68, no. 3, pp. 561–577, 2019. [Online]. Available: https://doi.org/10.1108/IJPPM-04-2018-0136

C. S. Byington, M. J. Roemer, and T. Galie, "Prognostic enhancements to diagnostic systems for improved condition-based maintenance [military aircraft]," in *Proceedings, IEEE Aerospace Conference*, vol. 6, March 2002, pp. 6–6.

A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *2008 International Conference on Prognostics and Health Management*, Oct 2008, pp. 1–9.

Z. Zhao, B. Liang, X. Wang, and W. Lu, "Remaining useful life prediction of aircraft engine based on degradation pattern learning," *Reliability Engineering & System Safety*, vol. 164, pp. 74 – 83, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0951832017302454

F. Lu, J. Wu, J. Huang, and X. Qiu, "Aircraft engine degradation prognostics based on logistic regression and novel os-elm algorithm," *Aerospace Science and Technology*, vol. 84, pp. 661 – 671, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1270963818308514

W. Yan, "Application of random forest to aircraft engine fault diagnosis," in *The Proceedings of the Multiconference on "Computational Engineering in Systems Applications"*, vol. 1, Oct 2006, pp. 468–475.

L. Liu, S. Wang, D. Liu, Y. Zhang, and Y. Peng, "Entropy-based sensor selection for condition monitoring and prognostics of aircraft engine," *Microelectronics Reliability*, vol. 55, no. 9, pp. 2092 – 2096, 2015, proceedings of the 26th European Symposium on Reliability of Electron Devices, Failure Physics and Analysis. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0026271415001687

and and and, "Remaining useful life prognostic estimation for aircraft subsystems or components: A review," in *IEEE 2011 10th International Conference on Electronic Measurement Instruments*, vol. 2, Aug 2011, pp. 94–98.

Y. Zhang, S. Ren, Y. Liu, and S. Si, "A big data analytics architecture for cleaner manufacturing and maintenance processes of complex products," *Journal of Cleaner Production*, vol. 142, pp. 626 – 641, 2017, special Volume on Improving natural resource management and human health to ensure sustainable societal development based upon insights gained from working within 'Big Data Environments'. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0959652616310198

E. Balmus, "Aircraft data acquisition," *INCAS Bulletin*, vol. 8, no. 1, p. 141, 2016.

A. Imani and M. Montazeri-Gh, "A multi-loop switching controller for aircraft gas turbine engine with stability proof," *International Journal of Control, Automation and Systems*, vol. 17, no. 6, pp. 1359–1368, 2019.

G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013, vol. 112.

S. Karsoliya, "Approximating number of hidden layer neurons in multiple hidden layer bpnn architecture," *International Journal of Engineering Trends and Technology*, vol. 3, no. 6, pp. 714–717, 2012.

S. Spielberg, A. Tulsyan, N. P. Lawrence, P. D. Loewen, and R. Bhushan Gopaluni, "Toward self-driving processes: A deep reinforcement learning approach to control," *AIChE Journal*, vol. 65, no. 10, p. e16689, 2019. [Online]. Available: https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.16689

G. Xiong, Z.-L. Li, Y. Ding, and L. Zhu, "Integration of optimized feedrate into an online adaptive force controller for robot milling," *The International Journal of Advanced Manufacturing Technology*, vol. 106, no. 3, pp. 1533–1542, 2020.

A. T. Abbas, N. Sharma, S. Anwar, F. H. Hashmi, M. Jamil, and H. Hegab, "Towards optimization of surface roughness and productivity aspects during high-speed machining of ti–6al–4v," *Materials*, vol. 12, no. 22, p. 3749, 2019.

F. Ridwan, X. Xu, and G. Liu, "A framework for machining optimisation based on step-nc," *Journal of Intelligent Manufacturing*, vol. 23, no. 3, pp. 423–441, 2012.

S. Stemmler, D. Abel, O. Adams, and F. Klocke, "Model predictive feed rate control for a milling machine," *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 11–16, 2016.

Y. Shaban, M. Aramesh, S. Yacout, M. Balazinski, H. Attia, and H. Kishawy, "Optimal replacement of tool during turning titanium metal matrix composites," in *Proceedings of the 2014 Industrial and Systems Engineering Research Conference*, 2014.

——, "Optimal replacement times for machining tool during turning titanium metal matrix composites under variable machining conditions," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 231, no. 6, pp. 924–932, 2017.

S. Ochella, M. Shafiee, and C. Sansom, "Adopting machine learning and condition monitoring pf curves in determining and prioritizing high-value assets for life extension," *Expert Systems with Applications*, vol. 176, p. 114897, 2021.

V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan, “Finding a "kneedle" in a haystack: Detecting knee points in system behavior,” in *2011 31st International Conference on Distributed Computing Systems Workshops*, 2011, pp. 166–171.

M. Shafto, M. Conroy, R. Doyle, E. Glaessgen, C. Kemp, J. LeMoigne, and L. Wang, “Modeling, simulation, information technology & processing roadmap,” *National Aeronautics and Space Administration*, vol. 32, no. 2012, pp. 1–38, 2012.

I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

T. Hagan Martin, B. Demuth Howard, H. Beale Mark *et al.*, “Neural network design,” *University of Colorado at Boulder*, 2002.

C. IEA, “Global ev outlook 2020,” *URL: https://www. iea. org/reports/global-ev-outlook-2020*, 2020.

M. Wolf, “Embedded software in crisis,” *Computer*, vol. 49, no. 1, pp. 88–90, 2016.

M. Ehsani, Y. Gao, S. Longo, and K. Ebrahimi, *Modern electric, hybrid electric, and fuel cell vehicles*. CRC press, 2018.

N. Ullah, M. Morisio, and A. Vetro, “A comparative analysis of software reliability growth models using defects data of closed and open source software,” in *2012 35th Annual IEEE Software Engineering Workshop*, 2012, pp. 187–192.

H. T. Hussein, M. Ammar, and M. M. Hassan, “Three phase induction motor’s stator turns fault analysis based on artificial intelligence,” *International Journal of System Dynamics Applications (IJSDA)*, vol. 6, no. 3, pp. 1–19, 2017.

M. E. H. Benbouzid, D. Diallo, and M. Zeraoulia, “Advanced fault-tolerant control of induction-motor drives for ev/hev traction applications: From conventional to modern and intelligent control techniques,” *IEEE transactions on vehicular technology*, vol. 56, no. 2, pp. 519–528, 2007.

H. Xue, D. Ding, Z. Zhang, M. Wu, and H. Wang, “A fuzzy system of operation safety assessment using multi-model linkage and multi-stage collaboration for in-wheel motor,” *IEEE Transactions on Fuzzy Systems*, pp. 1–1, 2021.

S. Kabir and Y. Papadopoulos, “A review of applications of fuzzy sets to safety and reliability engineering,” *International Journal of Approximate Reasoning*, vol. 100, pp. 29–55, 2018.

A. K. Jardine and A. H. Tsang, *Maintenance, replacement, and reliability: theory and applications*. CRC press, 2013.

H. A. Taha, A. H. Sakr, S. Yacout, and P. Serafin, "Failure reasoning and uncertainty analysis for wheel motor electric bus," in *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2021, pp. 1–4.

C. IEA, "Global ev outlook 2021," *URL: https://www.iea.org/reports/global-ev-outlook-2021*, 2021.

F. MACCIONI, "Maintenance program for electric vehicles power train by reliability centred maintenance," 2012.

D. Diallo, M. Benbouzid, and A. Makouf, "A fault-tolerant control architecture for induction motor drives in automotive applications," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 6, pp. 1847–1855, 2004.

A. Sangwongwanich and F. Blaabjerg, "Monte carlo simulation with incremental damage for reliability assessment of power electronics," *IEEE Transactions on Power Electronics*, vol. 36, no. 7, pp. 7366–7371, 2021.

E. Dehghan-Azad, S. Gadoue, D. Atkinson, H. Slater, P. Barrass, and F. Blaabjerg, "Sensorless control of im for limp-home mode ev applications," *IEEE Transactions on Power Electronics*, vol. 32, no. 9, pp. 7140–7150, 2017.

M. O. Sonnaillon, G. Bisheimer, C. De Angelo, and G. O. García, "Online sensorless induction motor temperature monitoring," *IEEE Transactions on Energy Conversion*, vol. 25, no. 2, pp. 273–280, 2010.

L. Samaranayake and S. Longo, "Degradation control for electric vehicle machines using nonlinear model predictive control," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 1, pp. 89–101, 2018.

D. Gonzalez-Jimenez, J. del Olmo, J. Poza, F. Garramiola, and P. Madina, "Data-driven fault diagnosis for electric drives: A review," *Sensors*, vol. 21, no. 12, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/12/4024

P. R. Thies, G. H. Smith, and L. Johanning, "Addressing failure rate uncertainties of marine energy converters," *Renewable Energy*, vol. 44, pp. 359–367, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960148112001437

C. Heising *et al.*, "Ieee recommended practice for the design of reliable industrial and commercial power systems," *IEEE Inc., New York*, 2007.

U. S. D. of Defense, *Reliability prediction of electronic equipment*. US Department of Defense, 1991.

J. H. Purba, D. Sony Tjahyani, A. S. Ekariansyah, and H. Tjahjono, "Fuzzy probability based fault tree analysis to propagate and quantify epistemic uncertainty," *Annals of Nuclear Energy*, vol. 85, pp. 1189–1199, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306454915004065

S. Kabir and Y. Papadopoulos, "A review of applications of fuzzy sets to safety and reliability engineering," *International Journal of Approximate Reasoning*, vol. 100, pp. 29–55, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888613X18301671

E. Zafiropoulos and E. Dialynas, "Reliability and cost optimization of electronic devices considering the component failure rate uncertainty," *Reliability Engineering System Safety*, vol. 84, no. 3, pp. 271–284, 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0951832003002734

S. Yacout, "Ind8217e- analytics of faults and maintenance," Fall 2022.

F. H. Gandoman, A. Ahmadi, P. V. den Bossche, J. Van Mierlo, N. Omar, A. E. Nezhad, H. Mavalizadeh, and C. Mayet, "Status and future perspectives of reliability assessment for electric vehicles," *Reliability Engineering System Safety*, vol. 183, pp. 1–16, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0951832018306215

S. Halder, A. Ghosal, and M. Conti, "Secure ota software updates in connected vehicles: A survey," *arXiv preprint arXiv:1904.00685*, 2019.

M. S. Hassan, A. A. Zaki Diab, M. Shoyama, and G. M. Dousoky, "Interleaved pwm strategy for common-mode leakage current and emi noise reduction of paralleled single-stage dc-ac converters," in *2020 IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2020, pp. 768–774.

H. Chun, S. Han, Jun, J. Lee, Lee, Jeong, Joo, and Jung, "Converter switching noise reduction for enhancing emc performance in hev and ev," in *PCIM Europe 2016; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management*, 2016, pp. 1–8.

Z. Liu and L. Zhang, "A review of failure modes, condition monitoring and fault diagnosis methods for large-scale wind turbine bearings," *Measurement*, vol. 149, p. 107002, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263224119308681

D. I. 15243, "Rolling bearings—damages and failures—terms characteristics and causes," 2004.

J. E. Brown, D. B. Harris, and F. G. King Jr, "Heavy-duty truck test cycles: combining driveability with realistic engine exercise," *International Journal of Heavy Vehicle Systems*, vol. 7, no. 4, pp. 299–316, 2000.

A. Traue, G. Book, W. Kirchgässner, and O. Wallscheid, "Toward a reinforcement learning environment toolbox for intelligent electric motor control," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 3, pp. 919–928, 2022.

APPENDIX A CHAPTER 3: SELF-HEALING MECHANISM INTERACTIONS

A.1 P2D is better than P2P in recovery time

While running the 20 sets sequentially that are labeled as out-of-specification, we compared the recovery time of the P2D and the P2P distance approaches. The P2D self-healing outperforms the P2P and recovers the out-of-specification state of the CNC machine faster than with P2P. This result was obtained when the initial settings were the $2^{nd}, 7^{th}, 8^{th}, and\ 9^{th}$ that are shown in Table 5.8 and Table 5.9. This behavior happened in 4-runs out of 20 runs, which led to out-of-specification as shown in Table 5.8. In 9 settings out of the 20, the recovery time was found to be the same for the P2D and the P2P approaches. In this case, the selection of a recovery pattern does not affect recovery time. Finally, seven settings led to a longer recovery time with the P2D than with the P2P. **A.1.1** $7^{th}, 8^{th}, and\ 9^{th}$ **Initial Settings**

These initial sets are different in feed rate and are common in 4×10^4 RPM spindle speed and with a 32 mm non-variable depth of cut. The different feed rate settings led to different faulty forces and temperature measurements for each initial set, as indicated in Table 5.8. In these runs, P2D-Self-healing outperforms the P2P-Self-healing, and it recovers the CNC machine Ra fault in 3.5 seconds as shown in Figures 6.a, 7.a, and 8.a. The P2D approach calculates the distance to each recovery pattern by considering it as uniform distribution. The P2P distance was calculated to a recovery pattern mean as a single point regardless of the other corrective values inside the pattern. P2D-Self-healing selects the 3^{rd} recovery pattern to be the nearest pattern in the first 5 cycles, which is the same as P2P. Then, it selects the 1^{st} recovery pattern instead of the 3^{rd} pattern at a time of 3 seconds, while the P2P-self-healing recovers the Ra out-of-specification in 18 seconds with the 1^{st} recovery pattern, as given in Figures A.1.b, A.2.b, and A.3.b. In these initial sets, the P2P-Self-healing cleared the detected fault with the corrective action of changing the initial settings to 499.3 mm/min feed rate and spindle speed to 3.56×10^4 RPM. Consequently, the fault detection symptoms of the force in the x-direction given in Figures A.1.d, A.2.d, and A.3.d, F_x were changed from the values that cause a fault in each run to a normal value of 18.3 N. P2D-self-healing corrective settings are 308.2 mm/min feed rate and 3.13×10^4 RPM spindle speed. As a result of changes to the settings, the faulty force in the x-direction for each initial set, F_x was changed to 15 N as presented by Figures A.1.d, A.2.d, and A.3.d, and the mean temperature decreased to 243.5 C^o, as given in Figures A.1.f, A.2.f, and A.3.f. **A.2 P2D and P2P have the same recovery time.**

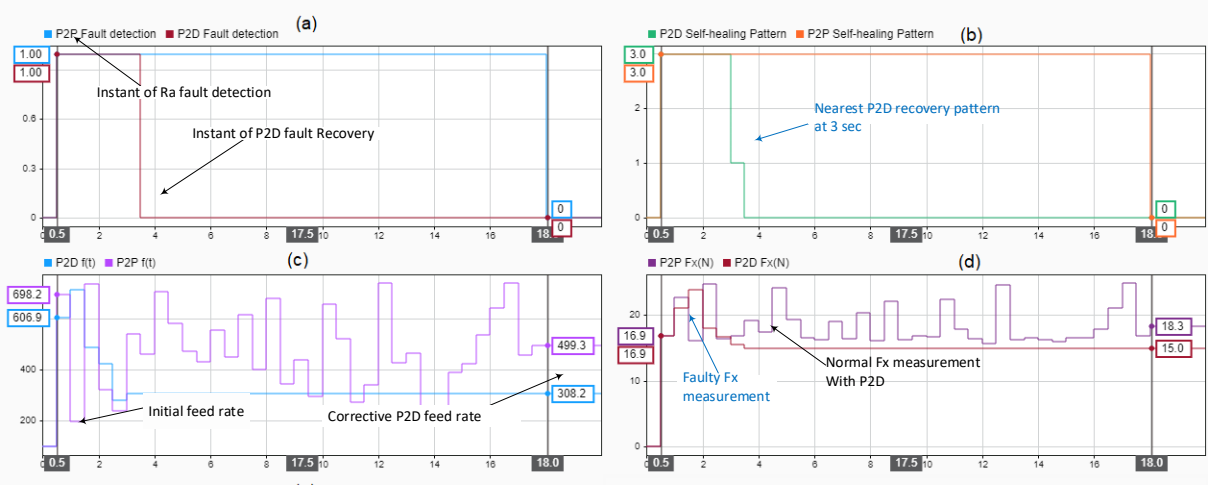


Figure A.1 P2P-synchronized Self-healing (P2P & P2D) interacting with CNC milling fault in the 7^{th} run

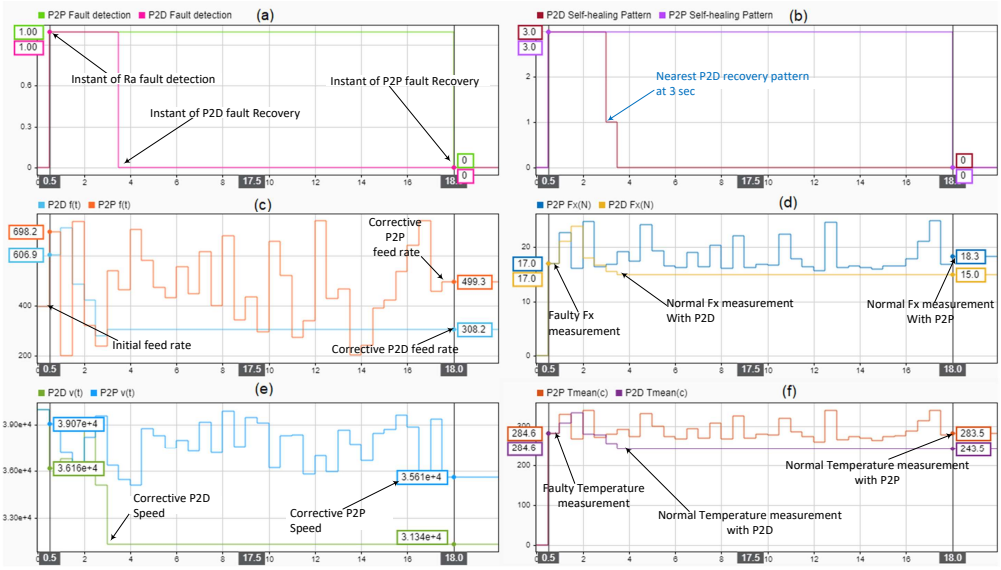


Figure A.2 Synchronized Self-healing (P2P & P2D) interacting with CNC milling fault in the 8^{th} run

In the runs of this behavior, the distance approaches of P2D and P2P do not affect Self-healing performance, and it has the same fault recovery performance for the two approaches. The performance equalizations exist in 9 runs out of the 27 runs that are indexed $[18^{th} - 27^{th}]$ with excluding the conformed settings 25^{th} initial set. **A.1.2** 18^{th} **Initial Settings**

In this run, we set the CNC machine initially with faulty settings of a non-variable 64 mm cutting depth, a feed rate of 700 mm/min, and $4{\times}10^4$ RPM spindle speed. P2P and P2D Self-healing select the 2^{nd} recovery pattern to be the nearest pattern to the 18^{th} initial set-

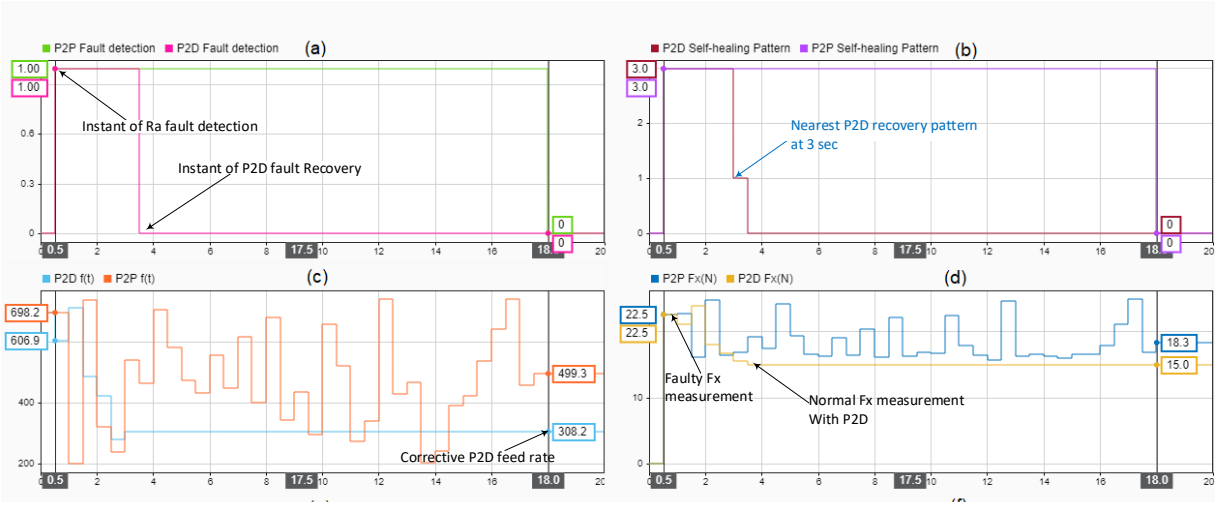


Figure A.3 Synchronized Self-healing (P2P & P2D) interacting with CNC milling fault in the 9^{th} run.

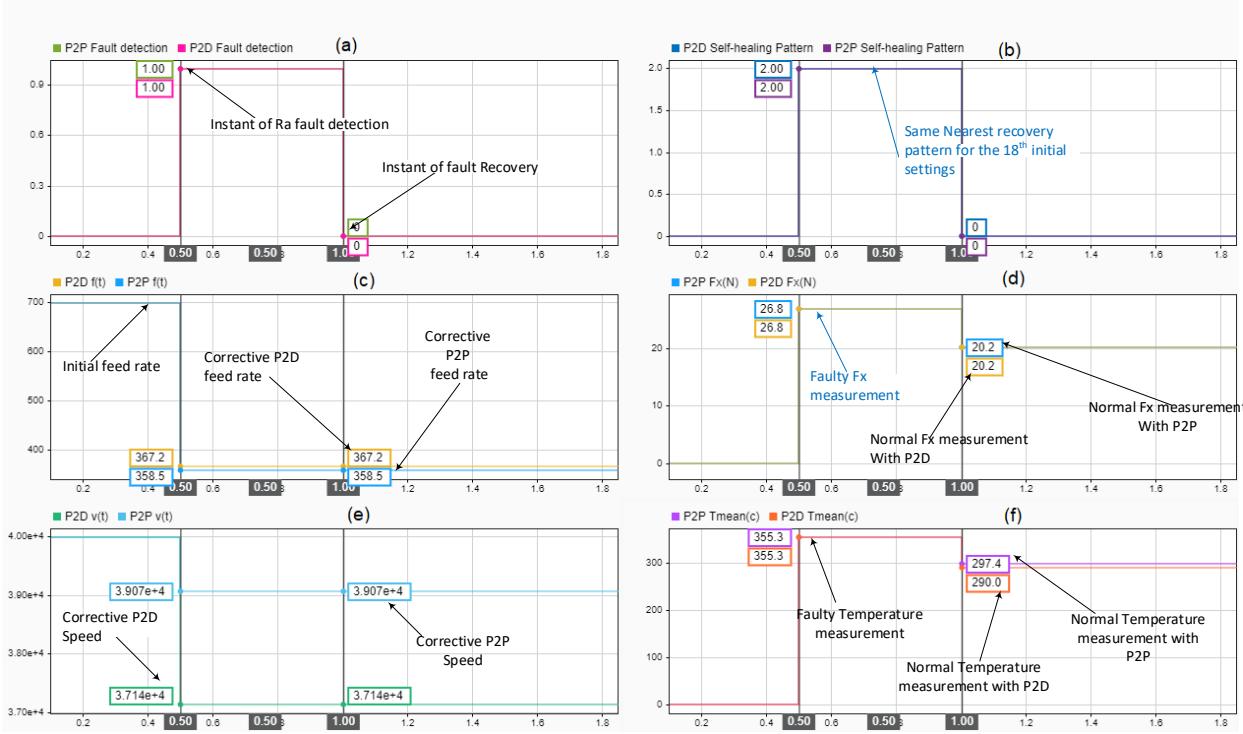


Figure A.4 Synchronized Self-healing (P2P & P2D) interacting with CNC milling fault in the 18^{th} run.

tings, as identified by figure A.4.b. The self-healing module achieves the same fault recovery time of 1 second for the two distance approaches, as given in figure A.4.a. The self-healing corrective actions are 358.5 mm/min, 3.9×10^4 RPM with P2P self-healing, while P2D self-

healing generates 367.2 mm/min and 3.7×10^4 RPM for the feed rate and speed, respectively. The fault cause was the F_x sensor that has a high value of 26.8 N at the incident of the fault, and with the self-healing action, it was reduced to normal values of 20.2 N, as depicted by figure A.4.d. The mean temperature decreased from 355.3 C^o to 297.4 C^o with P2P, and to 290 C^o with P2D, as presented in figure A.4.f. **A.1.3** 19^{th}**, and** 20^{th} **Initial Sets.**

The CNC machine's initial faulty settings are 96 mm non-variable depth of cut, 1×10^4 RPM spindle speed, and the feed rate is 100 mm/min for the 19^{th} set and 400 mm/min for the 20^{th} initial set. According to the depth of cut, the nearest possible recovery pattern to be selected is the 4^{th} pattern out of the recovery patterns in Table 5.6. Therefore, in these runs, the distance approaches do not influence the self-healing module performance, and it has the same self-healing corrective actions of 200.3 mm/min feed rate and 2.94×10^4 RPM spindle speed with P2P and P2D self-healing, as recorded by figure A.5.c and figure A.5.e. The Self-healing module clears the detected out-of-specification fault within 1.5 seconds, as given in figure A.5.a. As the initial feed rate is different in the 19^{th} and 20^{th} initial sets, the faulty measurements of forces and temperature are different for each initial set, as indicated in figure A.5.d and figure A.5.e. CNC milling x-access force was the main fault indicator in the 19^{th} and 20^{th} runs, and it decreased with self-healing actions to 24.0 N; while the mean temperature changed to 294.4 C^o, as exhibited in figure A.5.d and figure A.5.e.

In the 21^{st}, 22^{nd}, 23^{rd}, 24^{th}, 26^{th}, and 27^{th} runs, the self-healing with either P2P or P2D has the same fault recovery time, and it clears the CNC machine fault in 1.5 seconds, as mentioned in figure A.6.a. The depth of cut is the common setting in all these runs, and it is 96 mm; therefore, the 4^{th} recovery pattern is the nearest possible pattern as in the 20^{th} run. The self-healing actions to recover the Ra fault are the same values of 200.3 mm/min for the feed rate and 2.94×10^4 RPM for the spindle speed, as mentioned in figure A.6.c and figure A.6.e. For all these 6 runs, the F_x is the main cause of the Ra fault and it activated the 1^{st} IF-THEN detection rule in section 4.2. The faulty F_x sensor values were 69.9 N, 55.3 N, 41.8 N, 41.8 N, 40.7 N, 29.7N, and 28.1 N for the 21^{st}, 22^{nd}, 23^{rd}, 24^{th}, 26^{th}, and 27^{th} Runs respectively, and it changed with a self-healing module to 24 N, as shown by figure A.6.d.

A.3 P2P is better than P2D in recovery time

This is the last kind of Self-healing performance, and it was in 7 runs out of the 27 runs. However, P2D and P2P approaches select the same nearest recovery pattern in each run of these 7 runs; Self-healing P2P outperforms P2D self-healing, and it recovers the CNC machine Ra fault in fewer time steps than with P2D distance. In the 3^{rd}, 4^{th}, and 6^{th} runs, P2P and P2D Self-healing selects the 6^{th} recovery pattern to recover the out-of-specification fault, as presented in figure A.7.3.b, figure A.7.4.b, and figure A.7.6.b, respectively. The P2D self-healing recovers the Ra fault within 2.5 seconds, while the recovery time was 2 seconds

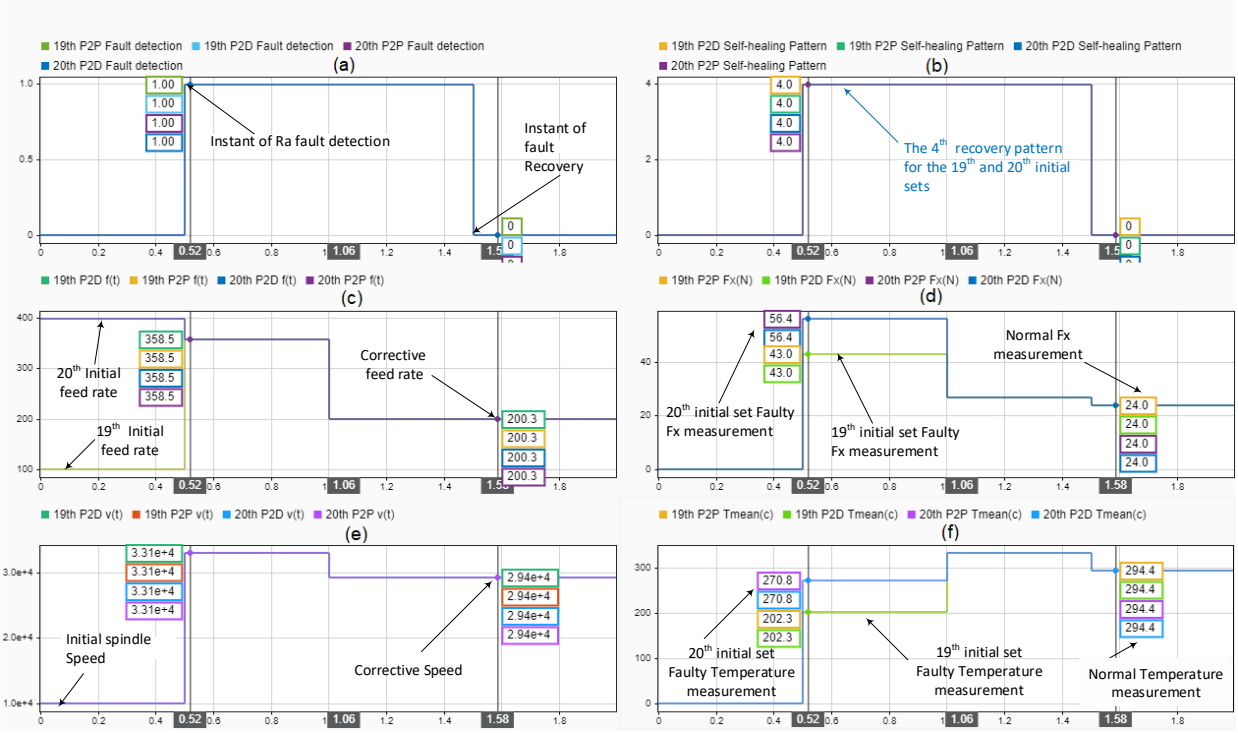


Figure A.5 Synchronized Self-healing (P2P &P2D) interacting with the CNC milling Ra fault in sets 19 and 20.

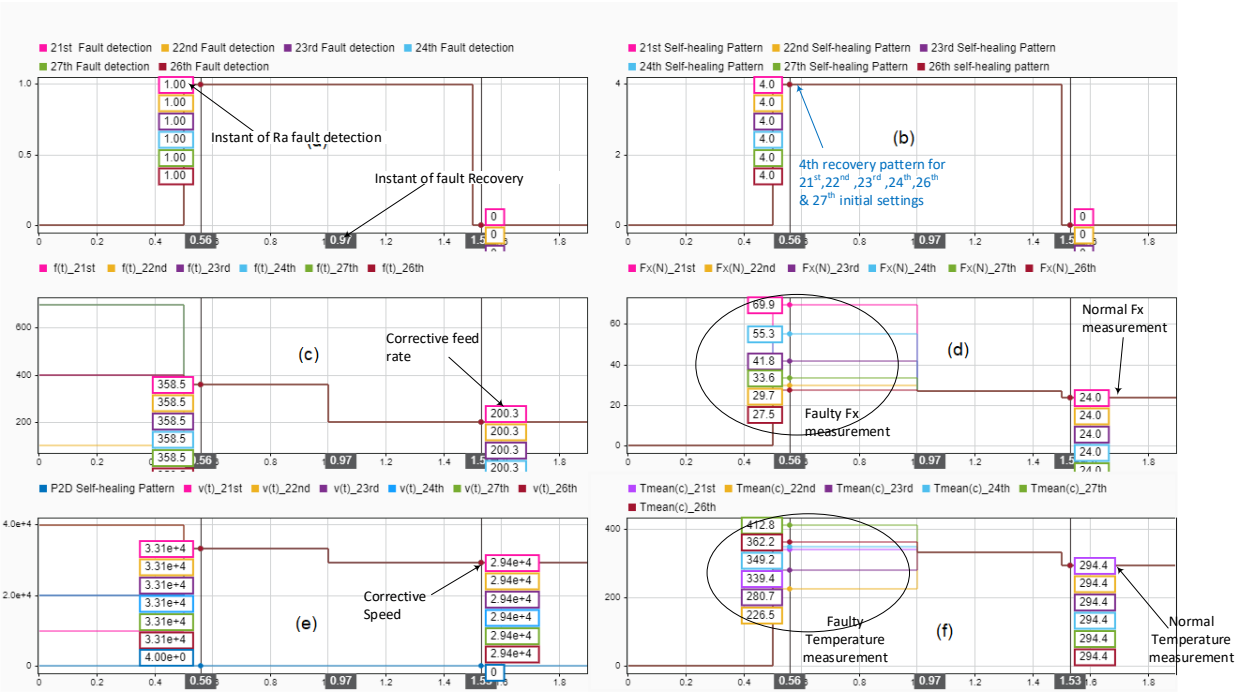


Figure A.6 Synchronized Self-healing interaction with Ra fault in the 21^{st}, 22^{nd}, 23^{rd}, 24^{th}, 26^{th}, and 27^{th} runs.

with P2P self-healing, as indicated in figure A.7.6. In the remaining 4 initial sets out of the 27, the 5^{th} recovery pattern is selected to be the nearest to clear the Ra faults in the 10^{th}, 11^{th}, 12^{th}, and 15^{th} runs, according to the P2P and P2D distance calculations. P2P Self-healing clears the CNC machine fault within 1 second, while it is cleared in 1.5 seconds with P2D, as given in figure A.8.

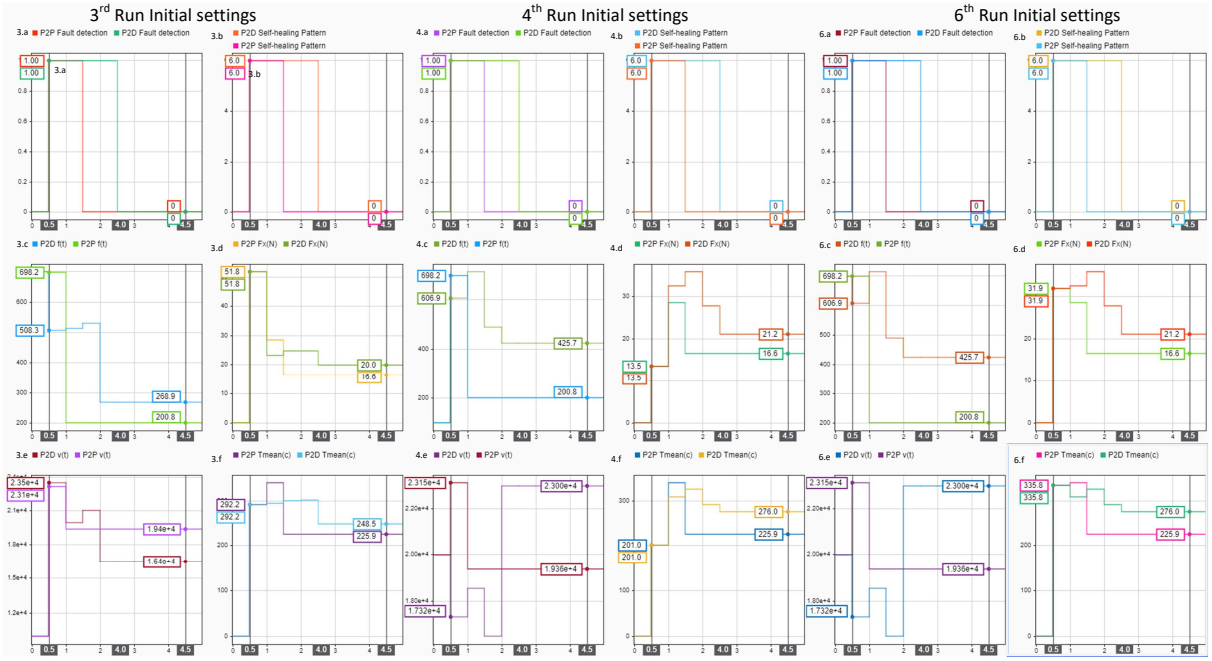


Figure A.7 Synchronized Self-healing (P2P & P2D) interacting with the CNC milling fault in the 3^{rd}, 4^{th}, and 6^{th} runs.

The P2D Self-healing minimum execution time was 1.9 mseconds and it is achieved at the 12^{th} run, while the maximum P2P execution time was 2 µseconds at the 15^{th} run, as stated in figure A.9. Hence, the P2D self-healing recovers the fault within the next fault detection cycle of 0.5 seconds later than P2P-self-healing. The execution time of Self-healing with P2D distance depends on the number of corrective sample points inside the recovery pattern, which are sampled from the pattern range to calculate the covariance matrix (Cov). While, P2D-self-healing would be more accurate with a higher number of samples, its exclusion time would increase. The number of recovery samples used in all 27 runs was 1000 samples. When we decrease the number of samples to 10 samples per recovery pattern, a P2D Self-healing execution becomes small and almost equal to the P2P-Self-healing execution time. Moreover, the fault recovery time gets faster, the same as with P2P self-healing, in these 7 runs as given in figure A.9.

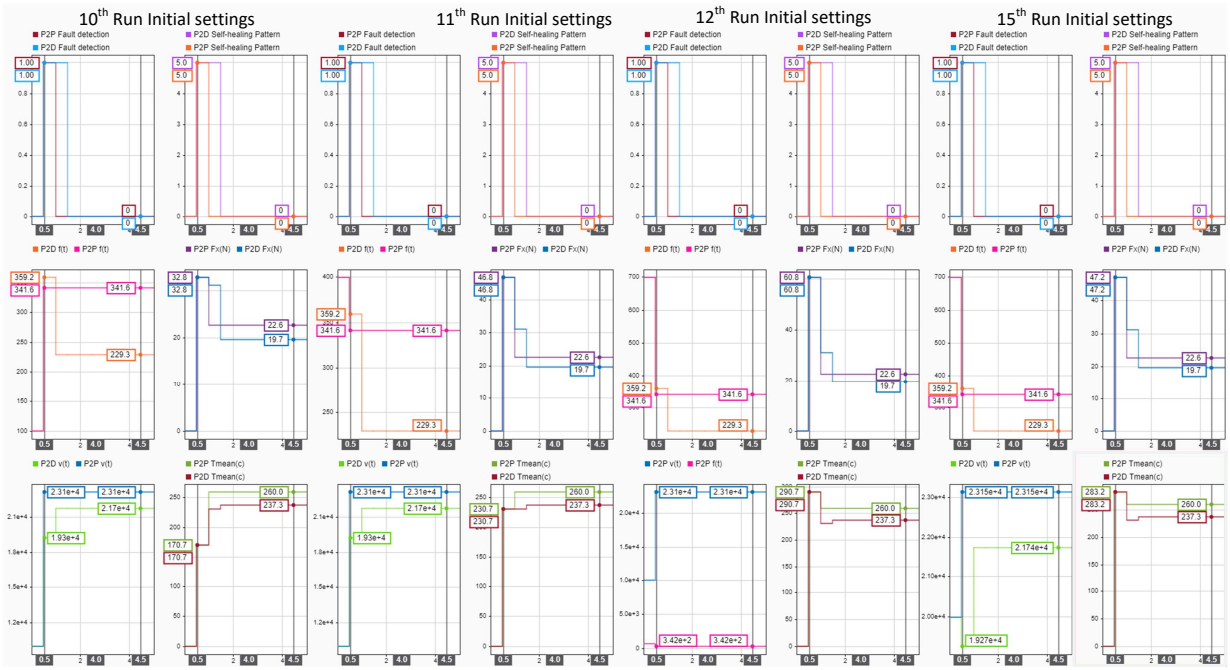


Figure A.8 Synchronized Self-healing (P2P &P2D) interacting with the CNC milling fault in the 10^{th}, 11^{th}, 12^{th}, and 15^{th} runs.

Figure A.9 Self-healing algorithm execution time with the3^{rd}, 4^{th}, 6^{th}, 10^{th}, 11^{th}, 12^{th}, and 15^{th} Initial Sets

APPENDIX B CHAPTER 5: PRE-FAILURE DRL AGENT

B.1 DRL Agent Parameters

The proposed pre-failure DDPG architecture consists of two deep actor-critic networks with two hidden layers and the hyperparameters in Table B.1. The DRL performance depends on its hyperparameters, and it is related to the environment/application dimension space of actions and state [33, 34, 89, 133]. These hyperparameters are adopted from literature on computer game applications that have the same data dimensions of actions and sensors as the CNC turning machine [34]. Figure B.1 shows the developed Pre-Failure agent architecture.

Table B.1 DRL Pre-Failure agent hyperparameters

Hidden neurons	Discount factor	Batch size	Learning rate critic	Learning rate actor	Target Update rate	Memory size
256/128	0.995	128	1e-4	1e-4	1e-3	1e6

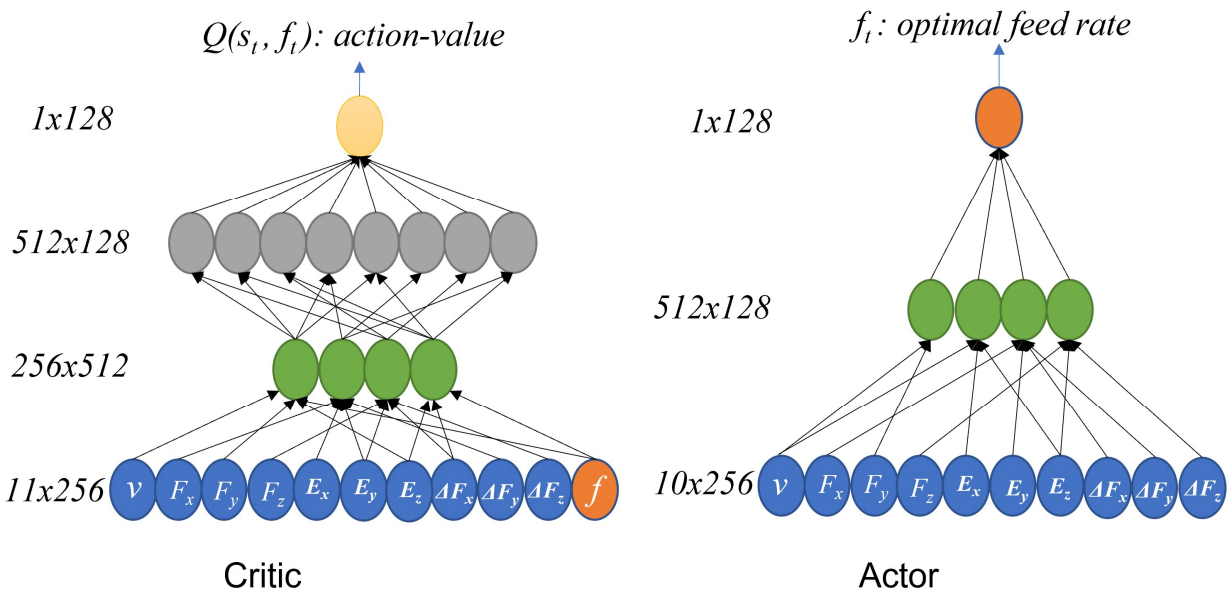


Figure B.1 Pre-failure DDPG agent architecture for CNC tool performance

B.2 DRL Interactions for Different Runs

In this section, the detailed results of runs II,III,IV, and V are stated.

B.2.1 Run II, the spindle speed is 7500 RPM

The Pre-Failure agent interaction in this run is demonstrated by figure B.2.

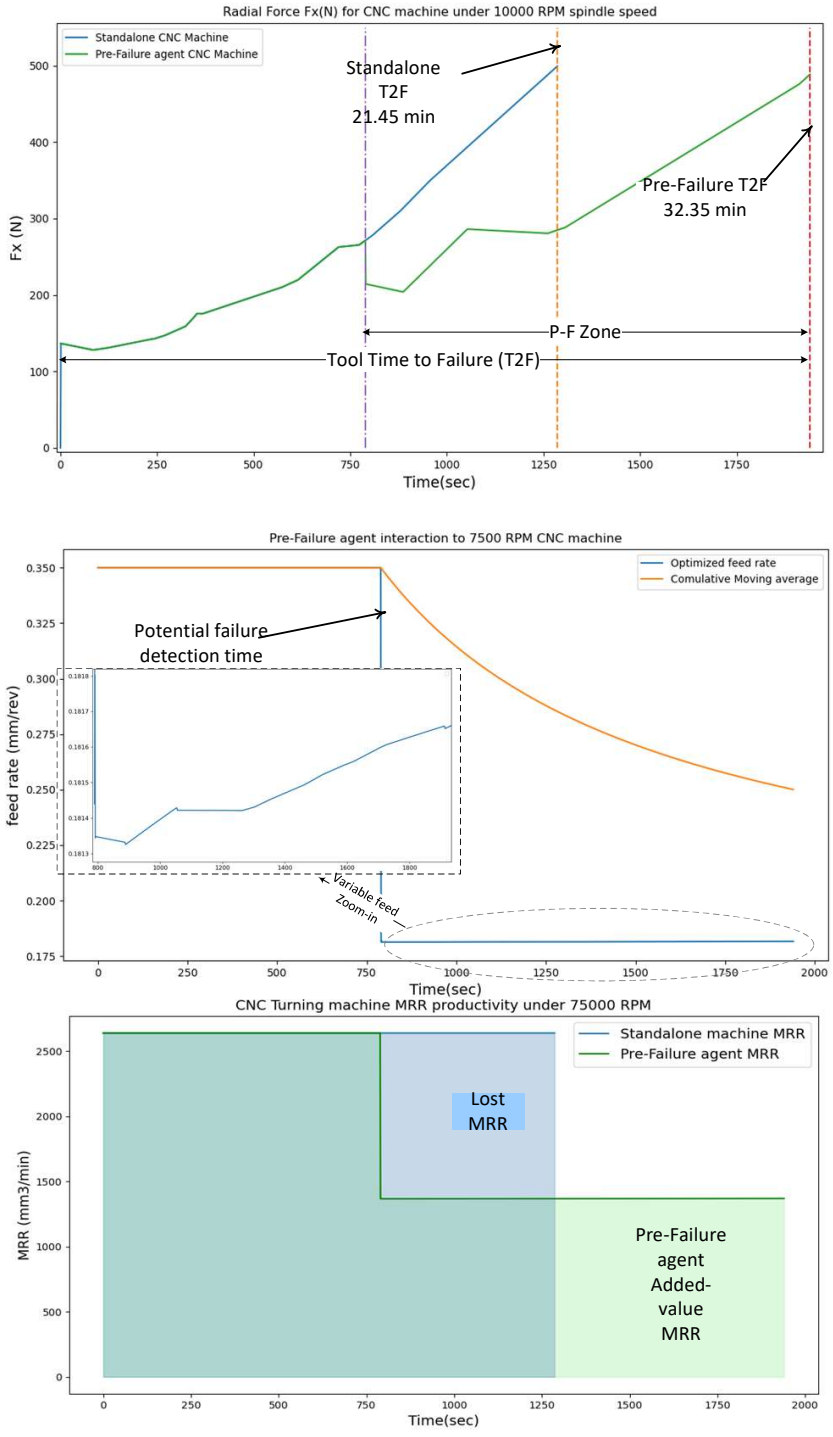


Figure B.2 Pre-Failure agent interaction in Run II, the spindle speed is 7500 RPM

B.2.2 In Run III, the spindle speed is 1000 RPM

The Pre-Failure agent interaction in this run is given by figure B.3.

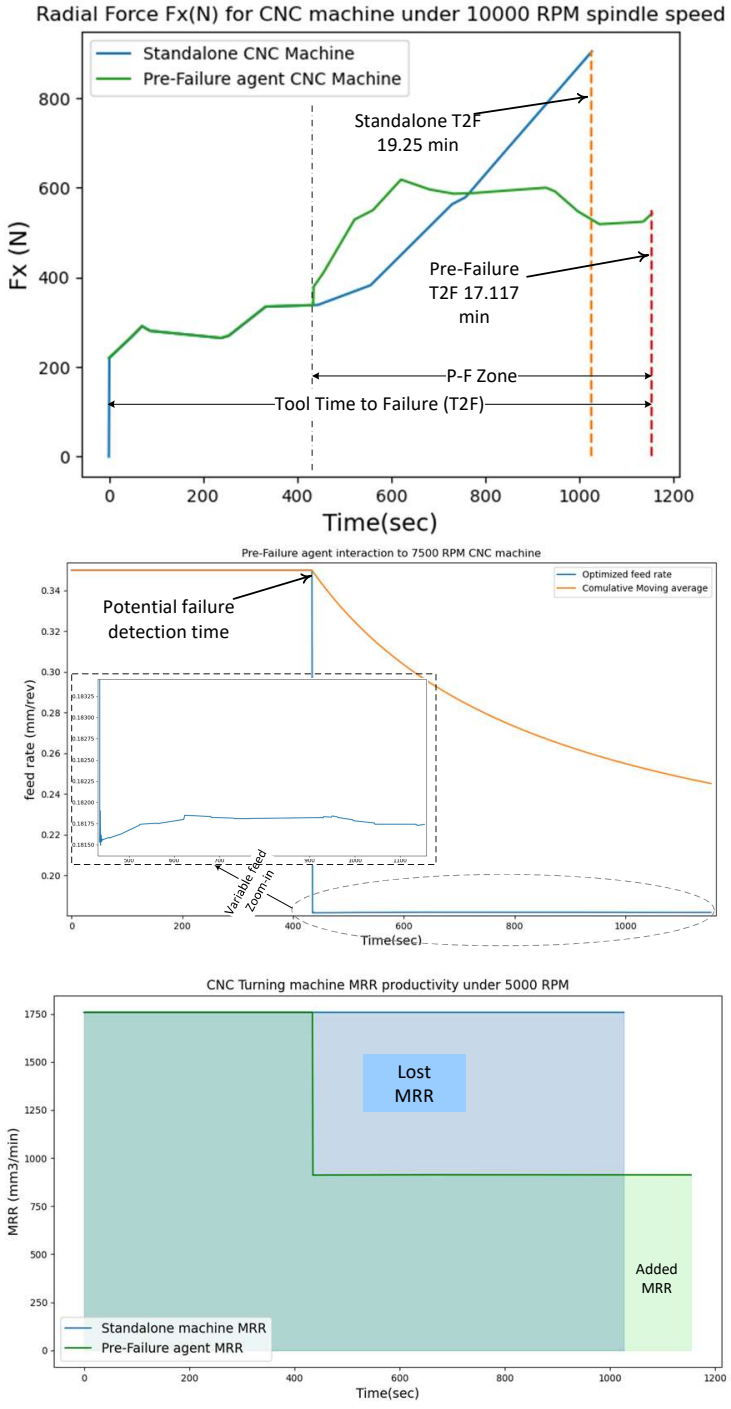


Figure B.3 Pre-Failure Agent Interaction in Run III, the spindle speed is 1000 RPM

B.2.3 In Run IV, spindle speed is 12,500 RPM

The Pre-Failure agent interaction in this run is depicted by figure B.4.

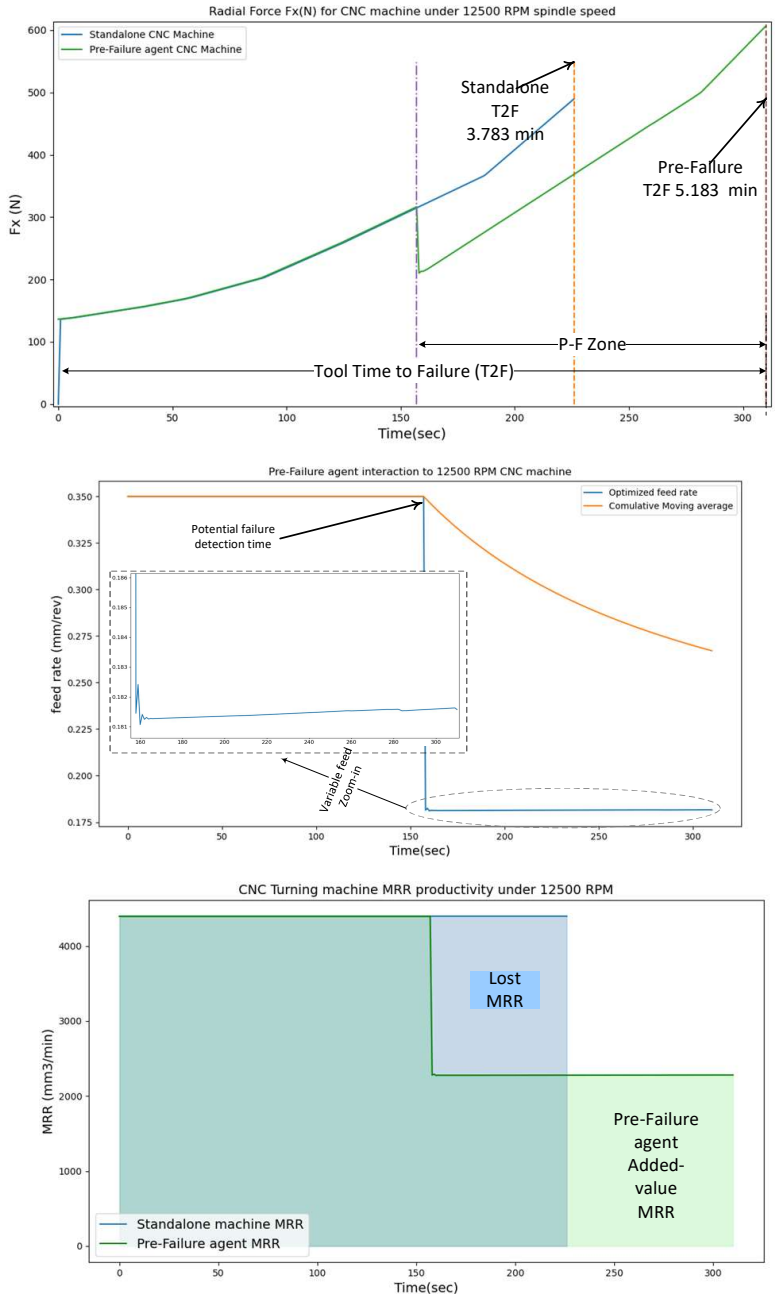


Figure B.4 Pre-Failure Agent Interaction in Run IV, spindle speed is 12,500 RPM

B.2.4 In Run V, the spindle speed is 15000 RPM

The Pre-Failure agent interaction in this run is shown by figure B.5.

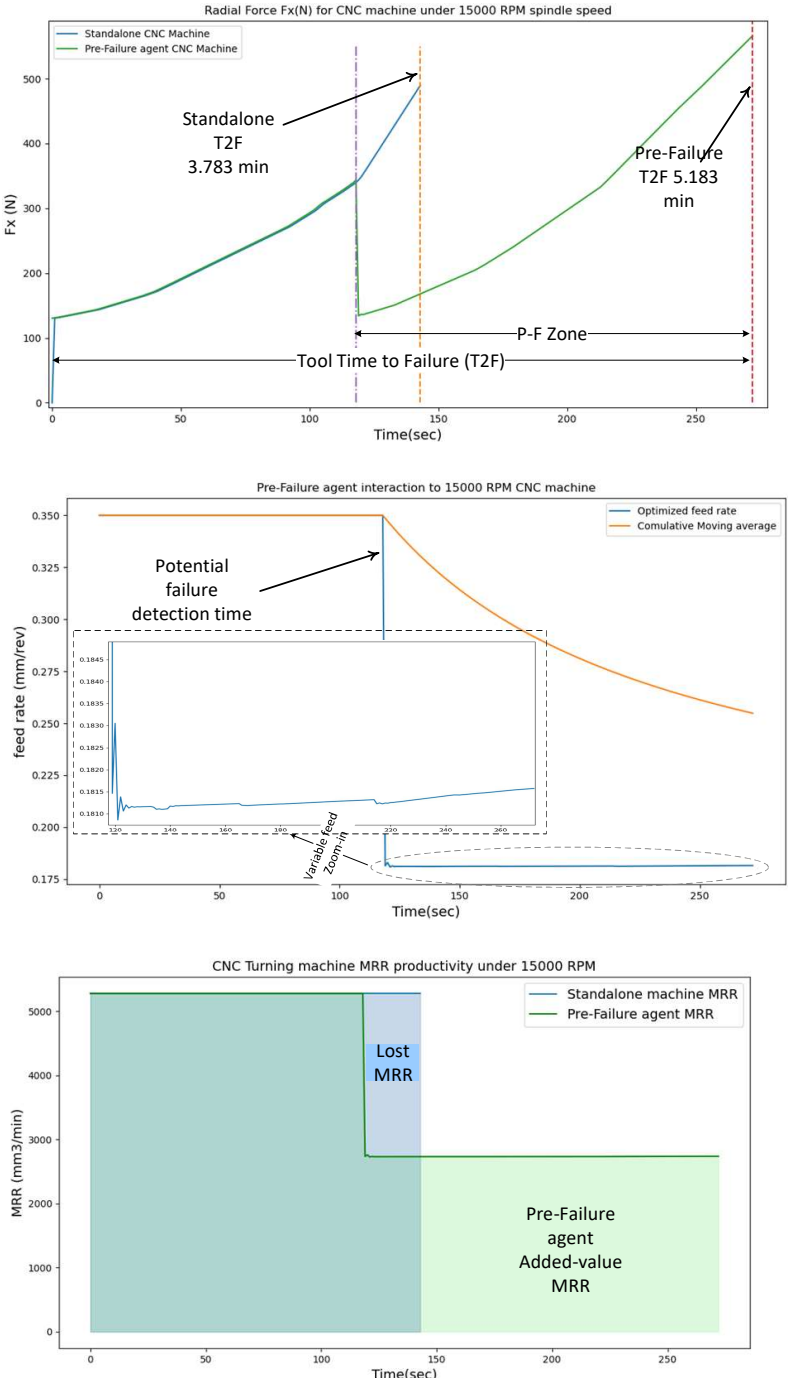


Figure B.5 Pre-Failure Agent Interaction in Run V, the spindle speed is 15000 RPM

APPENDIX C CHAPTER 7: W.M.E-BUSES' RCM ACTION LIST

Table C.1 Full RCM actions worksheet for W.M.E-Bus.

Inx	Sub-System	Failure Mode	Effects/consequence	possible cause	Detection	PM Task / healing-action	Frequent	Autonomic loop	Ref
C.1	ECU	ECU cannot initialize	E-Bus shutdown	Software bugs	Regular inspection	automatic check new software and update	daily	Yes	[31]
C.2	ECU	ECU cannot initialize	E-Bus shutdown	Over/under power voltage protection circuit SMD component	NA	Time-direct inspection	1-Year	NA	[6, 38]
I.1	3-Ph DC/AC Inverter	output Power failure	high vibration	IGBT failure (single limp)	Motor current	-Reconfigure the Inverter's switches	instant of failure	Yes	[10]
I.2	3-Ph DC/AC Inverter	Unintended motive power loss	Damages the components in the device	Over-current and overvoltage caused an excessive heat generated by the spike in current (or) voltage	Motor current	Time-direct inspection	2-years 30000 mil	NA	[25]
I.3	3-Ph DC/AC Inverter	Abnormal noise	Noise	Motor-inverter integration computability	Auditable Vibration	Check inverter frequency	each software upgrade	Yes	[32, 33]
I.4	3-Ph DC/AC Inverter	Improper installation of the Inverter	The vehicle could not start	-Sub-System integration does not follow the user manual - Improper cable selection -Inline fuse failure	Regular inspection	Time-direct inspection	2-years 30000 mil	NA	[6, 39]
S.1	Sensors	speed measurement	-No signal -vibration -abnormal readings -Motor out	- signal/CAN cable failure (Broken, Noise, Shield) -Connector failure(Rust, Broken) -Sensor Failure -Communication circuit failure	No speed signal Abnormal readings	-Sensorless Motor Control -Time-direct inspection	3-years 45000 mil	Yes	[9, 11, 34]
S.2	Sensors	Temperature measurement failure							[12]
B.1	Motor's Bearing	FLAKING/spalling	Materials are split off from the smooth surface of the raceway or rolling elements	- Excessive load - Entry of foreign debris, water penetration - Poor lubrication, - Unsuitable bearing clearance - Progression from rust -High/Low speed -Current leakage (pitting)	-Vibration sensor, if applicable	-Replace Bearing -Improve lubrication -Change Design -Derate Operating Speed, if applicable	2-year 22000 mil	Yes	[15, 35, 36, 40]
B.2	Motor's Bearing	SCORING	scratches	-Excessive load. -Poor lubrication -Particles are caught between surfaces with relative motion - Inclination of inner and outer rings -Shaft bending -Speed too slow -Rapid Change in rotation direction	-Vibration sensor, if applicable -Motor current sensor, if applicable	-Replace Bearing -Improve lubrication -Change Design -Check the operating speed -Check driving behavior	2-year 22000 mil	Yes	[15, 35, 36]
B.3	Motor's Bearing	PEELING	cloudy spots appear on the surface along with light wear	-Unsuitable lubricant -Entry of debris into the lubricant - Rough surface due to poor lubrication -Surface roughness of mating rolling part	-Vibration sensor, if applicable	-Replacement -Improve lubrication -Design change	2-year 22000 mil	NA	[35, 36]
B.4	Motor's Bearing	SMEARING	Surface damage, the appearance of nicks, grooves, smearing	-High speed and light load - Sudden acceleration/deceleration -Improper lubricant - Entry of water	-Vibration sensor, if applicable	-Replace Bearing -Improve lubrication -Change Design -Check the operating speed -Check driving behavior	2-year 22000 mil	Yes	[35, 36]

Table C.2 Full RCM actions worksheet for W.M.E-Bus (continue and end).

Inx	Sub-System	Failure Mode	Symptoms/consequence	possible cause	Detection	PM Task / healing-action	Frequent	Autonomic loop	Reference
B.5	Motor's Bearing	CAGE DAMAGE	Cage damage or deformation	-Poor mounting -Poor handling -Excessive rotation speed, sudden acceleration, and deceleration -Poor lubrication -Temperature rise	-Vibration sensor, if applicable -Motor current sensor, if applicable	-Replace Bearing -Improve lubrication -Change Design -Check the operating speed -Check driving behavior	2-year 22000 mil	Yes	[35, 36]
B.6	Motor's Bearing	SEIZURE	Melted/deformed raceway rings, rolling elements, and cage	-Excessive load -Excessive rotational speed -small internal clearance -Entry of water and debris -High speed -High acceleration & deceleration	Temperature sensor, if applicable	-Replace Bearing -Improve lubrication -Change Design -Check the operating speed -Check driving behavior	2-year 22000 mil	Yes	[35, 36]
B.7	Motor's Bearing	ELECTRICAL EROSION	-High-frequency Fluting occurs on the raceway surface	-Electrical potential difference between inner and outer rings -Electrical potential difference of a high frequency. -Motor earthing	Motor shaft to ground current	-Replacement -Prevent current flow through the bearings -Insulation of the bearing. -EMI filter & smooth inverter switching - Reducing Inverter switching frequency	2-year 22000 mil	Yes	[32, 40]
W.1	Motor's winding Sub-system	Stator Inter-turn fault	-Vibration, -Overheating -Melted winding	Internal temperature stress Insulation failure Oil/Water leakage	-Motor current, Temp sensor	-Monitoring the Degradation, if applicable - Derating at same load torque, if applicable -Regular inspection	1-year 30000 mil	Yes	[13, 15, 41]
W.2		Cage rotor broken bar							
W.3		wound rotor Inter-turn fault							
M.1	Mechanical link	Shaft	E-Bus shutdown	-Rotor failure -Misalignment	Vibration, if applicable	-Monitoring the Degradation, if applicable	1-year 15000 mil	NA	[6, 15]
M.2		Planetary gear breaking		-Fatigue -Friction between teeth -Misalignment		-Regular Inspection		Yes	