Time and Sensor Domain Data Decomposition and Analysis for Industrial Process Monitoring

A Novel Approach with Applications to Prognostic and Health Management of Cutting Tools

by

Yufei Gui

A dissertation submitted to

The University of Sheffield

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

8th April 2024

Yufei Gui

Time and Sensor Domain Data Decomposition and Analysis for Industrial Process Monitoring
A Novel Approach with Applications to Prognostic and Health Management of Cutting Tools
PhD Dissertation, 8th April 2024
Supervisor: Professor Zi-Qiang Lang

The University of Sheffield
Department of Automatic Control and Systems Engineering
Amy Johnson Building
Portobello Street
Sheffield, S1 3JD

Abstract

Industrial process monitoring aims to guarantee process safety and improve production efficiency by detecting the happening of faults or predicting the remaining useful life of key components. The data-driven process monitoring technique has been found quite useful across various fields, including chemical, semiconductor, and advanced manufacturing. However, the existing methods still face challenges including the lack of sufficient training data, the complicated time-varying change in process conditions, the redundancy of available signal features, etc. In this study, a novel framework referred to as time and sensor domain data decomposition and analysis, that has potential to be used to address these problems, is developed and validated in application to an essential topic in advanced manufacturing process monitoring, namely the prognostic and health management of cutting tools.

This dissertation contains seven chapters. In Chapter 1, the research background and objectives are described. A comprehensive review of the data-driven tool condition monitoring (TCM) methods are presented in Chapter 2. The novel time-sensor decomposition framework is established in Chapter 3, with a direct application to fault diagnosis of cutting tools in milling processes. Afterwards, three main topics including process anomaly detection, time-varying process monitoring, and process states prediction are studied, respectively. Chapter 4 proposes a time-sensor domain synthesis framework to resolve the problem of insufficient anomaly samples. Chapter 5 develops a recursive time-sensor decomposition approach to adapt to the time-varying trend of process variations. Chapter 6 introduces a local regularisation assisted split augmented Lagrangian shrinkage algorithm to deal with the feature selection problem in the existence of numerous redundant features. Both experiment and simulation studies are conducted to show the the performance of the proposed methods. In Chapter 7, some conclusions obtained during this project and future works are summarised. The prime contributions of this project are summarised as follows.

1. A novel time and sensor domain data decomposition and analysis framework is proposed, which compresses the multi-sensor signals into a lower number of time and sensor domain components (TDCs and SDCs) with dominant information remaining. Both time and sensor domain feature are used to build a tool fault diagnosis model. The results show that the obtained features grasp the dominant information and achieve a stable and optimal performance in different cases.

2. A novel time-sensor domain synthesis framework is designed to provide a solution to unsupervised detection of cutting tool breakage without the involvement of a training data set. The sensor domain information is exploited to determine the baseline cutting passes while the time domain information is used to generate tool condition-related features. The approach can create an unsupervised tool breakage detection model, and adaptively update the threshold to realise accurate anomaly detection.

3. A novel recursive time-sensor decomposition algorithm is proposed by exploiting the temporal dependency among different data snapshots. The obtained SDCs and TDCs can represent the general trend and the overall time-varying behaviours (including both normal and fault-induced variations) of an industrial process, respectively. This enables the creation of novel TDCs-based control chart statistics, and SDCs-based adaptive control limits, ensuring a good trade-off between fault detection rate and false alarm rate.

4. A local regularisation assisted split augmented Lagrangian shrinkage algorithmis proposed under the Bayesian evidence framework, introducing an individual penalty parameter for each model coefficient. During the optimisation process, redundant variables can be pruned accordingly, which significantly reduces the overall computing complexity. In feature selection of process prediction problems, this algorithm method exhibits impressive model sparsity, prediction accuracy, and higher computational efficiency.

5. Extensive milling experiments have been conducted over the period of this project. To validate the effectiveness of the proposed algorithms in real manufacturing processes, all the experimental conditions are designed according to production specifications. During the experiments, common signals used in TCM are collected from the numerical control system and external sensors. The analysis results show that the challenges existing in current process monitoring systems can be effectively addressed under the proposed frameworks.

Acknowledgement

I would like to express my sincere gratitude to my supervisor, Prof. Zi-Qiang Lang, for his great support and supervision of my PhD project. He has provided me with many precious suggestions on but not limited to systematic thinking, method development, and academic writing. His understanding and support encouraged me to overcome difficulties on my road towards a PhD, especially during the COVID-19 pandemic.

I also want to thank the staff of ACSE and AMRC for providing me with such a world-leading research environment and acknowledge the support of the Engineering Faculty Scholarship during past years.

Moreover, many thanks to Hatim Laalej, Yunpeng Zhu, Zepeng Liu, Mengxiao Wang, and Tong Liu, for their help in both research and life at Sheffield.

Finally, I would like to dedicate this dissertation to my parents and my fiancée, Hang, for their selfless love. They will always be the lights that guide me home.

Contents

1

Introduction

1.1 Background and motivation

Industrial process monitoring plays a pivotal role in guaranteeing process safety and improving production efficiency. This has been found quite useful across various fields, including chemical, semiconductor, and advanced manufacturing [1–3]. Recent advances in measurement technology have led to a surge of industrial data, paving the way for data-driven approaches that leverage statistical analysis of multiple variables [4, 5]. Taking the tool condition monitoring (TCM) as an example, this section introduces the background and motivation of this project as follows.

In advanced manufacturing, metal machining is widely used to produce parts meeting the design requirements. The pursuit of high productivity/quality, and low time/economic cost is an eternal theme of the manufacturers. The machining process consists of a series of physical processes, which are generally analysed by investigating mechanical, thermal, and chemical mechanisms separately or synthetically. Among various machining elements, such as the machine tool, fixture, and workpiece, the cutting tool is identified as the most active element [6]. Because the tool is associated with material removal directly and the change in tool condition, including tool wear and breakage, significantly affects the surface integrity of the parts. In terms of the production cost, the cutter-related cost should never be ignored, too. To be specific, 3-12% of the total cost is used for the purchase of new cutters and 7-20% of machining downtime results from tool failure or tool replacement [7, 8].

Even though an unbroken tool is essential for perfect surface roughness and dimension accuracy, a tool inevitably damages itself during the cutting process due to mechanical abrasion and chemical effects [9]. There are two types of performance degradation of the tool, namely progressive wear and sudden breakage. According to ISO-8688-1, tool condition is determined by the width of tool flank wear at the cutting edge. A wear amount of 0.3 mm is suggested to be used as the tool life limit. Figure 1.1 shows the tool flank wear versus cutting time under progressive wear and tool breakage, respectively. Gradual wear indicates a typical tool degradation process, which can be divided into initial, normal and severe wear regions sequentially (see Figure 1.1 (a)). The tool wear increases at different rates in these regions. However, the tool breakage can happen suddenly and randomly at any stages as shown in Figure 1.1 (b). In this case, the tool wear amount may change abruptly. The tool breakage can be induced by many factors, such as the existence of hard particles within the workpiece, the intermittent cutting process, and the high-speed and heavy-load conditions. All these factors put extremely high stress and shocks over the tool. Minor breakage occurs in the form of cutting edge breakage like chipping. Catastrophic breakage could be breakage of the tool shank.

When the tool is severely worn or tool breakage occurs, the tool will lose its cutting ability. Experimental results show that using a tool in failure state could cause unexpected vibration and deteriorate the finished surface [10]. Ideally, the cutter should to be replaced when the flank wear width is as close to the critical value as possible or in the first place when breakage happens so that the utilisation of the cutting tool is maximised without damaging the parts. But the literature has pointed out that only 50-80% of the tool lifecycle is effectively used in practice [11].

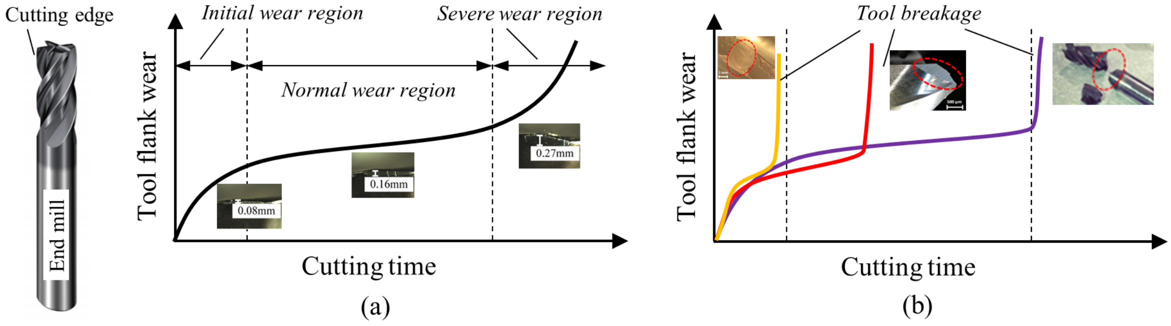


Figure 1.1: Schematic diagram of tool wear and breakage failure: (a) progressive tool wear, (b) sudden tool breakage occurring randomly during the cutting process.

To resolve this issue and realise prognostic and health management (PHM) of cutting tools, it is ideal to know the real-time condition of the tool during the machining process. The TCM technology can achieve this target. As mentioned in [10], an accurate and reliable TCM system can increase cutting efficiency and reduce overall costs by 10-50% and 10-40%, respectively. The TCM system is usually categorised into direct and indirect methods. Direct TCM method refers to directly measuring the changes in the cutting edge shapes by optical equipment and machine vision algorithms. From the image of the tool flank surface, the accurate tool condition can be quantified, such as the flank wear width, wear volume, and the degree of breakage. However, the direct TCM system cannot be exploited in real-time because the tool is constantly rotating and cutting the workpiece. Stopping the operation of the machine tool to measure the tool condition brings about additional downtime cost, which is not an optimal solution to the current problem.

Recently, intelligent manufacturing is reshaping the industry. Sensors are embedded into the manufacturing systems to provide the so-called industrial big data [8]. Now the indirect monitoring method enabling real-time monitoring of tool conditions has become the research hotspot. Physical signals, including spindle current and power, vibration, cutting force, sound, and acoustic emission (AE) are associated with the tool condition and can be acquired by different sensors. These signals reflect characteristics of the machining process. From them, tool condition-related information can be extracted using appropriate signal analysis algorithms. To analyse the sensor signals effectively, model-based and data-driven approaches are well developed in the literature.

The model-based approach identifies the tool condition using physical model of the cutting process [9, 12–14]. The model is constructed based on the principle in material mechanics and thermodynamics, which reveals the relationship between physical variables (e.g., force, vibration, power) in a machining process. The relationship itself, is related to the tool condition. Thus useful indices can be obtained by the model parameters evaluated from the sensor measurements. As such a model represents physically meaningful dynamics of the cutting process, it has the potential to be generalised to similar processes easily. However, the derivation of the physical model involves many theoretical assumptions. Sometimes the complex and nonlinear relationship in real world is significantly simplified such that the built model is not sensitive enough to the change in tool conditions. Hence, the model-based approach can only find limited applications in practice.

The drawback with the model-based approach can be overcome by the data-driven approach. Instead of building physical models, the data-driven TCM system directly utilises the relationship between the tool condition and sensor signals. It usually comprises four steps which are signal collection and pre-processing, feature extraction, feature reduction, and decision making. As mentioned above, a large volume of sensor data are now available in intelligent manufacturing. Given that multi-sensor signals describe the machining process more comprehensively, more studies are focusing on the multi-sensor data-driven approach by integrating different sensors such as the dynamometer, accelerometer, and AE sensor [15, 16]. Then the signal processing algorithms are performed to extract signal features in the time domain, frequency domain, and time-frequency domain [17]. For instance, common features include signal statistics, frequency spectrum, and time-frequency spectrum. These multi-sensor signal features consist of a matrix known as feature dictionary. However, using redundant features will not only increase computational complexity, but also introduce the risk of overfitting [18, 19]. It is necessary to develop a dimension reduction algorithm to obtain the most important features and avoid information redundancy. The important features can be a subset of all features or different combinations of original features. In terms of the decision-making step, many models can be used for tool breakage detection and tool wear monitoring. Nowadays, machine learning (ML) and deep learning (DL) models are widely investigated in this field [20–22]. To obtain better generalization performance, most of the learning models should be trained on a large training dataset. However, there is often a lack of sufficient abnormal samples for a machining process. Therefore, some researchers have paid attention to imbalanced data learning methods [23] and unsupervised learning models [24, 25].

This project tries to develop a process monitoring system based on multi-sensor data analysis. The performance of the proposed methods will be validated and demonstrated by applications to the PHM of cutting tools. Instead of focusing on the feature-layer information, this study aims to investigate a novel data-layer fusion strategy, which is still a gap in the literature [26]. This method not only provides a solution to the data compression problem in a monitoring system with a huge number of sensors, but also reveals the the interaction between original sensor signals in process monitoring.

The research idea is inspired by the concept of time-space decomposition in distributed parameter systems (DPSs) modelling [27, 28]. DPSs are usually time-space coupling and widespread in science and engineering. The system dynamics is typically modelled by partial differential equations (PDEs), which are difficult to solve directly. Researchers have proposed the so-called time-space separation modelling approach, which decomposes the time/space coupling kernels into separate time and space domains. By selecting a few dominant spatial basis functions (SBFs), the original system can be represented by a less number of time coefficients and SBFs, significantly reducing the model complexity.

In data-driven modelling, the Karhunen-Loéve decomposition (KLD) algorithm is widely used to determine empirical SBFs from the sensor measurements [28, 29]. This idea can be extended to the multi-sensor data analysis in process monitoring. Note that in DPSs modelling, sensors of the same type are deployed in different locations to measure one physical variable. After the time-space decomposition, the time coefficients represent the change in system dynamics over time while the SBFs are only dependent on spatial locations and described by data vectors. Unlike in DPSs modelling, the data in the process monitoring field are usually collected from heterogeneous sensors. These sensors are not necessarily located in specific locations as the spatial association is not of the main interest. Mathematically, the decomposition of heterogeneous signals is the same as in the case of homogeneous signals. The generated time coefficients, which are named time domain components (TDCs) in this study, still represent the time-varying properties of the process. While the basis functions no longer represent different spatial location-related information but the information related to different sensors. This domain is, therefore, referred to as the sensor domain and the basis functions are called sensor domain components (SDCs) in the present study. Consequently, the novel decomposition is named as time-sensor decomposition (TSD).

1.2 Challenges and objectives

This research focuses on the requirements of industrial process monitoring with special attention to PHM of cutting tools in advanced manufacturing. The main challenges that will be addressed are summarised as follows.

1. Framework design: even though the time-space decomposition approach has been well established in DPSs modelling, this study is the first attempt to develop a time and sensor domain data decomposition and analysis framework for multi-sensor data-driven process monitoring. The prerequisites that multi-sensor signals should meet remain unclear. The methodology for extracting different information from time and sensor domain components and integrating these information effectively has not been proposed. Except for the framework design, its advantages and limitations in real applications also need to be investigated systematically.

2. Process anomaly detection: in the machining process monitoring, one target is to detect the happening of tool breakage. Usually, a large amount of anomaly samples are necessary to determine an appropriate threshold or train a robust model. However, it is unrealistic to perform extensive destructive experiments to

obtain sufficient data sets. Even though the unsupervised method avoids model training, it suffers from the problem of baseline state determination. The empirical knowledge is not always reliable and sometimes fails to detect actual tool breakages.

3. Time-varying process monitoring: the changes of process states typically happen with a time-varying trend of variations. For example, the tool wear amount is always increasing, making it hard to find out when the tool wear amount exceeds a limit value. Many existing process monitoring models are too sensitive such that the normal tool wear increase rather than severe wear can trigger an alarm. Existing adaptive models rely on strict assumptions on the process itself, suffering from high false alarm rate, too. The challenge in the time-varying process monitoring is how to follow the slow and normal changes and trigger alarms only when a truly abnormal shift occurs.

4. Feature selection in process prediction: accurate prediction of process states forms the basis of process optimisation and control. For instance, estimating the remaining useful life (RUL) of the tool can avoid significant economic loss. However, multiple signals and numerous features may give rise to high computation cost and overfitting risk. It is necessary to discard redundant features to build a sparse regression model. Current methods cannot reach a good trade-off between model sparsity and prediction accuracy.

To fill these gap, this study, for the first time, proposes the time-sensor domain data decomposition and analysis framework and apply this method to tool condition monitoring. The following objectives will be investigated.

1. To systematically and comprehensively review the state-of-art of the TCM field and appraise the mainstream methodologies regarding both pros and cons, which provides a reference for researchers to better understand the key issues. The problem existing in practical applications will be analysed in order to propose more reliable monitoring methods.

2. To propose the methodology based on the novel idea of the time and sensor domain decomposition and associated feature extraction for process fault diagnosis models and make a comparison with raw signal features and feature-layer fusion methods.

3. To develop an unsupervised process anomaly detection method by synthesising the

obtained time and sensor domain information. The dependency on prior knowledge and extensive labelled data sets should be avoided.

4. To improve the time-sensor decomposition method so as to be applicable in time-varying process monitoring. Instead of applying TSD to each data snapshot separately, the temporal dependency among different data snapshots should be considered. A recursive TSD method should be developed to separate the general trend from the overall time-varying behaviours of an industrial process.

5. To develop a new feature selection algorithm such that redundant terms are removed and a sparse process states prediction model can be obtained. The proposed algorithm should exhibit a good trade-off among model sparsity, prediction accuracy, and computational efficiency.

6. To apply the proposed methods to real-world scenarios including tool fault diagnosis, tool breakage detection, tool wear monitoring, and tool wear prediction. It is expected that, in the future, the proposed TSD-based TCM methods can be implemented in practice as a part of the intelligent manufacturing system. Such a system will help manufacturers utilise their cutting assets more effectively and avoid serious sequences induced by cutting tool failure.

1.3 Research contributions

The prime contributions gathered throughout the development of this project are summarised as follows.

1. A novel time and sensor domain data decomposition and analysis framework is proposed. Under this framework, the multi-sensor signals are compressed into a lower number of time and sensor domain components with dominant information remaining. The analysis of TDCs and SDCs is investigated. Both time and sensor domain feature are used to build a tool fault diagnosis model, whose performance is demonstrated through milling experiments. The results show that the obtained features grasp the dominant information including the correlation relationship of raw signals, which help to achieve a stable and optimal classification performance in different cases.

2. A novel time-sensor domain synthesis framework is designed to provide a solu-

tion to unsupervised detection of cutting tool breakage without the involvement of a training data set. The sensor domain information is exploited to determine the baseline passes and adaptively update the TBD threshold while the time domain information is used to generate features for TBD. The approach can resolve difficulties with lacking sufficient data to train a robust supervised classification model for TBD, and circumvent problems with existing unsupervised TBD techniques in threshold determination.

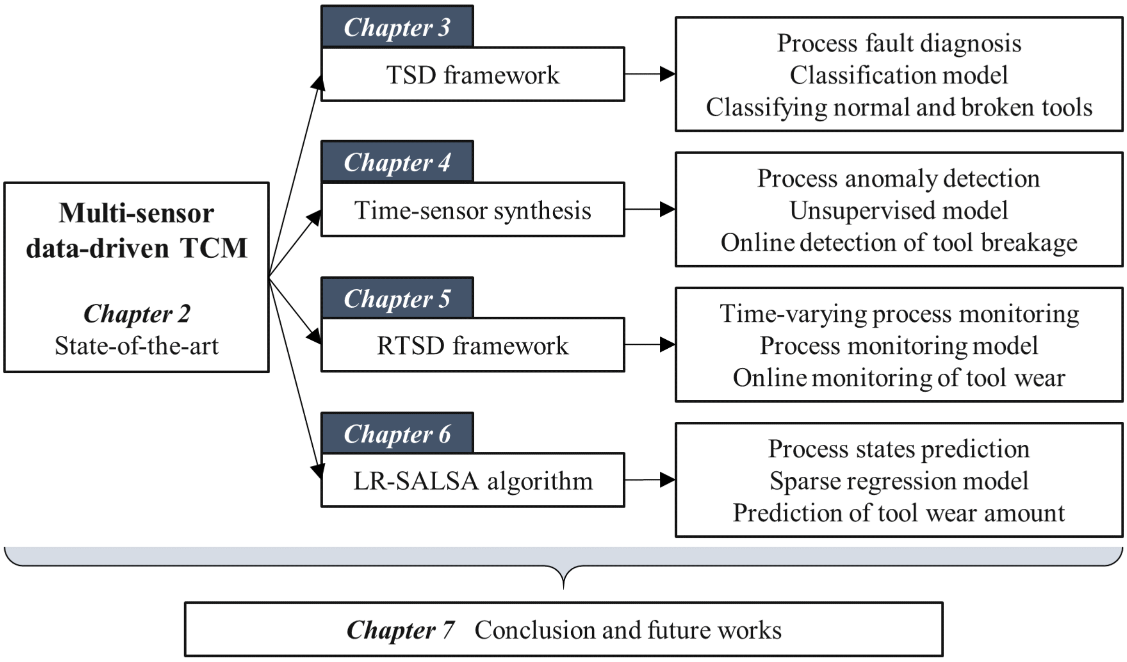
3. Furthermore, a novel recursive time-sensor decomposition algorithm is proposed by exploiting the temporal dependency among different data snapshots. The obtained SDCs and TDCs can represent the general trend and the overall time-varying behaviours (including both normal and fault-induced variations) of an industrial process, respectively. This enables the creation of novel TDCs-based control chart statistics, and SDCs-based adaptive control limits, ensuring a good trade-off between fault detection rate and false alarm rate. The proposed framework is applicable to time-varying industrial process monitoring.

4. A local regularisation assisted split augmented Lagrangian shrinkage algorithm (LR-SALSA) is proposed under the Bayesian evidence framework, introducing an individual penalty parameter for each model coefficient. An iterative algorithm is derived to optimise these hyperparameters relying on the dataset only. At each iteration step, according to the values of the penalty parameters, redundant variables can be pruned in time, which significantly reduces the overall computing complexity. In feature selection of process prediction problems, the LR-SALSA method exhibits impressive model sparsity, prediction accuracy, and higher computational efficiency.

5. Extensive milling experiments have been conducted over the period of this project. To validate the effectiveness of the proposed algorithms in real manufacturing processes, all the experimental conditions are designed according to production specifications. Various types of tool failure including breakage and excessive wear have been produced, which enables a comprehensive study on PHM of cutting tools. During the experiments, common signals used in TCM are collected from the numerical control (NC) system and external sensors. The proposed algorithms are then applied to analyse these multi-sensor data. Results show that the challenges existing in current TCM systems can be effectively addressed under the novel framework.

1.4 Dissertation outline

This thesis consists of seven chapters covering introduction, literature review, the proposed methods with applications to PHM of cutting tools, and conclusions. The layout of the thesis is shown in Figure 1.2.



Multi-sensor data-driven TCM
Chapter 2
State-of-the-art
Chapter 3
TSD framework
Process fault diagnosis
Classification model
Classifying normal and broken tools
Chapter 4
Time-sensor synthesis
Process anomaly detection
Unsupervised model
Online detection of tool breakage
Chapter 5
RTSD framework
Time-varying process monitoring
Process monitoring model
Online monitoring of tool wear
Chapter 6
LR-SALSA algorithm
Process states prediction
Sparse regression model
Prediction of tool wear amount
Chapter 7 Conclusion and future works

Figure 1.2: The layout of the rest of the thesis.

- Chapter 1: The research background, objectives, and contributions are described briefly in this chapter.

- Chapter 2: A comprehensive and systematic literature review of the data-driven TCM methods is presented, including the key steps of multi-sensor system, feature extraction, feature reduction, and model building. Most of the techniques are also applicable to industrial process monitoring.

- Chapter 3: A brief summary of the time-space decomposition method in DPSs modelling is given as a foundation of the proposed time-sensor decomposition framework. The specific methodology regarding the decomposition and analysis procedure is proposed, with a fault diagnosis model presented. The benefit of this novel framework, compared with raw signal feature-based method, is demonstrated through two case studies where the time and sensor domain components are directly applied to tool fault diagnosis in milling processes.

- Chapter 4: The problems with existing process anomaly detection methods are analysed first. Then a novel unsupervised model is proposed based on an innovative time-sensor domain synthesis framework. Based on the decomposition results, representative features are extracted from the TDCs. Meanwhile, the baseline stages and corresponding threshold can be determined and updated adaptively according to the clustering analysis of SDCs. The Mahalanobis distance-based method is then proposed to perform the anomaly detection. The proposed approach has been applied to the analysis of sensor data from two practical machining experiments for tool breakage detection. The results show the effectiveness and advantages of the new approach over existing techniques in terms of both computation time and overall accuracy in anomaly detection.

- Chapter 5: The challenges existing in time-varying process monitoring are reviewed. A novel time-varying process monitoring approach is proposed. Instead of applying TSD to each data snapshot separately, the influence of previous snapshots on the current multi-sensor data is considered in a recursive way. In this sense, the TSD algorithm is further extended to a RTSD method. After the data decomposition, the time and sensor domain components are better related to the overall changes and underlying trend of the process, respectively, which makes it applicable to time-varying process monitoring. The proposed approach has been validated through a simulation study and an experimental study on the tool wear process during dynamic milling. The results have demonstrated that the new approach can provide the best trade-off between fault detection rate and false alarm rate.

- Chapter 6: The importance of feature selection in process states prediction is pointed out first. Then a novel LR-SALSA algorithm is proposed for feature selection. An individual penalty parameter is introduced to each model coefficient such that redundant terms are removed immediately. An iterative algorithm, based on the Bayesian evidence framework, is derived to optimise the local regularisation hyperparameters automatically and guarantee model sparsity. The effectiveness of the proposed approach is demonstrated through both simulation and experimental studies. The results show that only LR-SALSA has a good trade-off among model sparsity, estimation accuracy, and computational efficiency in comparison to five existing feature selection methods.

- Chapter 7: Conclusions and future works are included in this chapter.

1.5 List of publications

- Yufei Gui, Zi-Qiang Lang, Zepeng Liu and Hatim Laalej. 'Recursive Time-sensor Domain Data Decomposition: A Novel Method for Time-varying Process Monitoring'. Submitted to: *Control Engineering Practice*.

- Yufei Gui, Zi-Qiang Lang, Zepeng Liu and Hatim Laalej. 'Tool Condition Monitoring Based on Nonlinear Output Frequency Response Functions and Multivariate Control Chart'. In: *Journal of Dynamics, Monitoring and Diagnostics* 2.4 (Nov. 2023), pp. 243-51.

- Yufei Gui, Zi-Qiang Lang, Zepeng Liu, Yunpeng Zhu, Hatim Laalej and David Curtis. 'Unsupervised Detection of Tool Breakage: A Novel Approach Based on Time and Sensor Domain Data Analysis'. In *IEEE Transactions on Instrumentation and Measurement* 72 (July 2023), pp. 1-13.

- Yufei Gui, Zi-Qiang Lang, Zepeng Liu, Yunpeng Zhu and Hatim Laalej. 'Time-Sensor Domain Data Decomposition and Analysis for Fault Diagnosis of Cutting Tools'. In: *2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. Dec. 2022, pp. 187-192.

2

Review of industrial process monitoring techniques with applications to PHM of cutting tools

2.1 Introduction

Nowadays, intelligent manufacturing has become an indispensable technology that shapes the future of industrial production. The UK government proposed the High Value Manufacturing Catapult in 2013 [30]. The catapult pointed out that the intelligent system was critical to increase the global competitiveness of products, and the industrial big data played an important role in advanced manufacturing. The German government described the combination of Internet technologies and future-oriented smart manufacturing as the '4th industrial revolution', which is also known as Industry 4.0 [31]. The core concept of this strategy was the Cyber-physical System (CPS). The condition of the virtual model of physical equipment in the CPS could be driven by process parameters. The intelligent sensing and monitoring technologies formed the foundation of a CPS.

It should be noted that the terminology of intelligent manufacturing is broad as manufacturing usually involves various equipment and techniques. But the metal machining process is still the mainstream in production. The importance of real-time tool condition monitoring during metal machining has been clarified in Chapter 1. The monitoring system of cutting tools and processes has received widespread attention from both industry and academia.

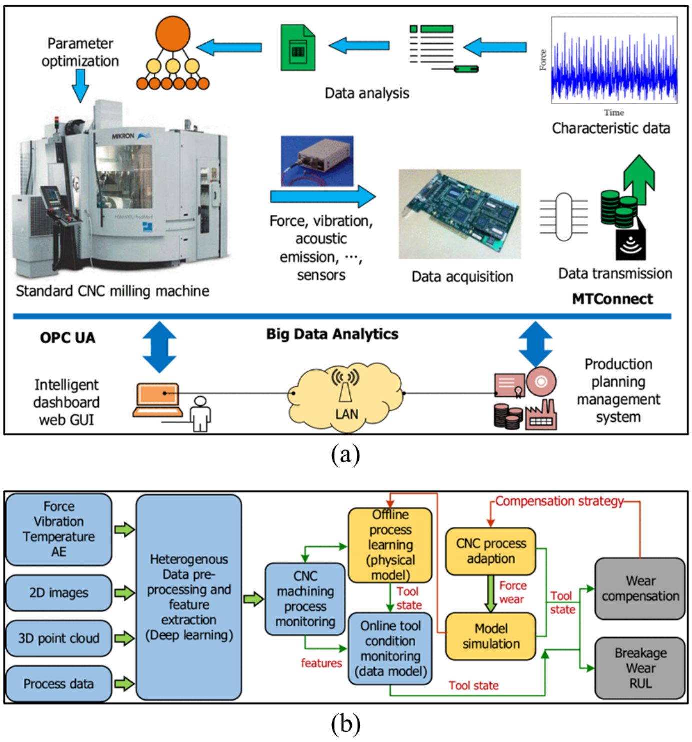


Figure 2.1: (a) Common monitoring system architecture; (b) Big data oriented tool condition monitoring framework [32].

Many commercial process monitoring products are available in the market. One of the key functions of these software is TCM. Figure 2.1 shows the architecture and framework of common TCM systems. The ARTIS GmbH has designed the CTM and GENIOR systems, and both offer flexible data capture methods [33]. The digital power data are obtained from the Numerical Controller (NC) and a wide variety of signals, such as cutting force, torque, and vibration, can be captured via external sensors. Based on the analysis of these data, the systems can detect tool breakage, tool wear and missing tools. The systems are also equipped with an adaptive control option allowing automatic learning of the alarm limits to ensure timely termination of the machine tool when activating the alarm. Furthermore, the ARTIS GENIOR system can be used for fully unmanned industrial configuration. The Sandvik Coromant group has released the CoroPlus Process Control modular [34]. It is also capable of collecting data from the NC system and external sensors. The system includes two solutions namely collision detector and tool guard. The former can detect the collision between the spindle and workpiece, and the latter can monitor the tool condition in real-time. If the tool breakage or missing tool is identified, the machining will be stopped automatically. The Caron

Table 2.1: Accessibility to multi-sensor signals in some commercial TCM systems

System	Power/Torque (NC)	Torque (External)	Force	Vibration	AE	Strain	Coolant
ARTIS	✓	✓	✓	✓	✓	✓	✓
CoroPlus Process Control	✓		✓		✓		
TMAC	✓		✓	✓	✓		
NORDMANN	✓		✓	✓	✓		
MONRONIX	✓		✓	✓	✓		
ToolScope	✓			✓	✓		

Engineering's real-time tool wear and breakage detection system, TMAC, also has multiple integrated sensors for condition monitoring [35]. Besides the tool wear detection, TMAC can automatically and instantaneously adjust machining conditions to optimise the process and reduce cycle time. Many other systems, including NORDMANN [36], MONTRONIX [37], and ToolScope [38], also provide connectivity to multi-sensor signals and basic functions like condition monitoring, adaptive feed rate control, and process deviation identification. Note that most of these systems realise process monitoring by simply comparing the real-time signal value with a threshold. The difference lies in the connectivity to user interfaces, accessibility to multiple sensors, and data analysis methods. Table 2.1 shows the signals used in these systems.

On the academic side, a large number of papers aiming to establish a TCM system have been published. Recent progresses in the TCM area are reviewed from different angles. For example, Rehorn et al. [10] presented a review of monitoring methods and results according to different machining methods, including end milling, face milling, drilling, and turning. Zhu et al. [39] investigated the state-of-the-art of a specific method, wavelet analysis, for TCM and further discussed future studies in this track. Zhou and Xue [8] organised their review following the basic process flow of a TCM system: sensor configuration, feature extraction, and monitoring model. Serin et al. [40] provided an overview of the literature on TCM and the most recent deep learning algorithms and pointed out new opportunities for applying deep learning methods in the TCM area. Recently, the tool breakage monitoring techniques have been comprehensively reviewed in terms of signal acquisition, feature extraction, and decision-making [6, 41].

In this chapter, representative literature relate to TCM methods is covered. This comprehensive review hopes to present the target and recent trend of four key steps in a typical TCM system, which are extensively used in the commercial system and reported by the literature over two decades. As illustrated in Figure 2.2, the four critical procedures pave the way from the machining process to the tool replacement decision.

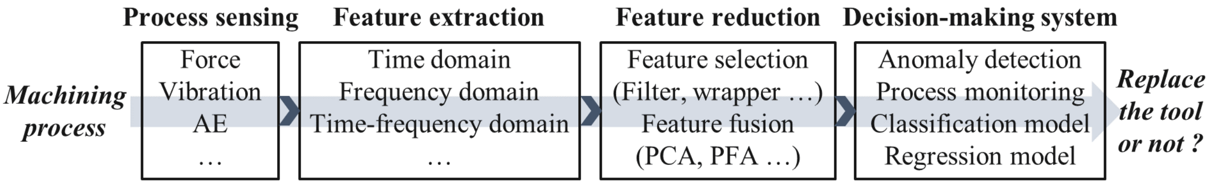


Figure 2.2: Four key steps of a TCM system.

2.2 Multi-sensor system

Process sensing technology is the foundation of the monitoring of a cutting process. The measurements of physical variables, such as the force induced by material removal, the vibration on the spindle and worktable, and the motor power of the machine tool, are collected by internal or external sensors. These sensors are usually mounted inside the machine tool and close to the cutting area. Because the sensors might have different frequency response property, the physical mechanisms behind the measurements are not the same. Thus, the multi-sensor configuration is helpful to improve the monitoring accuracy and reduce the sensitivity to environmental noise compared with the single-sensor scheme. This section will introduce the characteristics and applications of some popular sensor types.

2.2.1 Cutting force

Cutting force is an important variable in machining because it can guide the design of cutters and the selection of machining conditions. The force signal is typically measured by the dynamometer mounting on the worktable. When the tool becomes blunt, the friction between the cutting edge and workpiece will increase, leading to a larger cutting force. Many studies have shown that the tangential force is more sensitive to the tool wear and surface roughness and the axial force is associated with tool breakage [42, 43]. Huang et al. [44] designed a linear model with cutting force measurements as the input. They showed the model was effective for tool wear detection and fault diagnosis. Gao et al. [45] proposed a statistical analysis framework to extract latent and sensitive features from the cutting force signal. High monitoring accuracy was achieved using the proposed method.

Despite the benefits of cutting force-based monitoring, several limitations still exist in the deployment of dynamometers. As explained by Prickett and Johns [42], the physical size of the workpiece cannot be too large, otherwise, a plate-mounted sensor could not capture the slight variations of cutting force. Besides, the price of the dynamometer is quite high, which further limits its application in a TCM system.

2.2.2 Vibration

During a continuous machining process, the rotation speed of the cutting tool or workpiece is usually constant and induces periodic change, which is discernible through vibration signals. In condition monitoring, the vibration might be the most widely-used measurement due to the following advantages. First of all, the accelerometer is much cheaper and can be easily installed on different components including the spindle, workpiece, and fixture. Besides, the accelerometer has a high range of frequency response, enabling the detection of more details and a distinct separation between signals corresponding to sharp and dull cutting edges. This is attributed to the increased number of sliding contacts with the workpiece surfaces as the tool undergoes wear [46]. The utilisation of the vibration signal in TCM tasks has been studied in depth. For example, Yesilyurt and Ozturk [47] investigated the performance of different features obtained from the vibration signal in terms of revealing transient and progressive faults of the end mills. They found that the frequency-related features had a clear correlation with the tool wear. Kang et al. [48] validated the effectiveness of the peak period of spindle vibrations in tool chipping detection. The representative feature showed a significant increase as chipping occurred. Xie et al. [49] integrated a wireless vibration sensor directly into the tool holder in order to reduce the distance from the sensor to the cutting area. The integrated design simplified the acquisition of vibration signals.

Note that the high dynamic range of accelerometers is a double-edged sword. The vibration signal is not only sensitive to the cutting process but also to the cutting-irrelevant events (e.g. unbalance of the rotating parts, kinematic inaccuracy of the drives, the use of coolant) and environmental noises [50]. In this case, it is still a challenge to extract useful information from the raw vibration signal.

2.2.3 Acoustic emission

Acoustic emission is defined as the occurrence of exceptionally high-frequency oscillations resulting from the transient discharge of strain energy [42]. During the material removal process, the piezoelectric AE transducer can detect the signal associated with various phenomena, including plastic deformation of the material, friction stemming from cutter-workpiece interaction, and instances of tool breakage. The effective frequency range of an AE sensor is from 20 kHz to 2 MHz. This range is much higher than the mechanical vibration and environmental noises. Researchers have developed many AE signal analysis methods to monitor the tool failure and predict the amount of tool wear. Belgassim and Jemielniak [51] investigated the distribution moments of the AE signal collected from the turning process. The results showed that the catastrophic failure of the cutting tool could be precisely detected. Bhuiyan et al. [52] examined the frequency spectrum of the AE signals. The subcomponents of tool wear was separated from the parts related to material removal. A relationship between the AE signal and tool condition had been found.

The AE signal is quite special due to its insensitivity to low-frequency interference. This property improves the accuracy of tool breakage detection. However, besides the tooth chipping, the entrance and exit of the cutting edge from the workpiece also cause similar spikes in AE signals [42]. Hence, valid information of the tool condition is hard to be extracted exactly. In the literature, distinct AE signal processing algorithms have been proposed to address specific processes, thereby precluding the establishment of a universal analysis procedure. In additional, the placement of the AE sensor has a strong influence on the signals availability. The AE signal-based TCM system is found to be more feasible in low-load machining like finishing or drilling with small-diameter cutters [53].

2.2.4 Motor current or power

The motor current or power refers to the electric energy that drives the linear motion of axial systems and the rotary motion of the spindle. These signals are in proportion to the driving force and torque and thus can be regarded as indirect measurements of the cutting force. The amplitudes will increase with the evolution of the tool wear. The current can be measured by a Hall sensor and the power is computed by the product of the current and voltage. According to the literature, the motor current/power is highly recommended because of the low cost and there is no need to adjust the machining zone. Li et al. [54] built a model to estimate the feed cutting force according to the AC servomotor current measurements. Then the force estimates were used to determine the onset of rapid tool wear. Drouillet et al [55] utilised the root mean square (RMS) of the spindle power to predict the RUL of the cutting tool. Experimental results showed that only using power signal could still obtain a fair prediction accuracy. Shen et al. [56] also extracted a useful feature from the spindle power signal using time-frequency analysis. The index showed a high correlation coefficient with the tool wear measurement.

Table 2.2: Accessibility to multi-sensor signals in some commercial TCM systems

Sensor	Signal	Pros	Cons
Dynamometer	Force	Sensitive to the cutting process Straightforward analysis	Complicated installation High economic cost
Accelerometer	Vibration	Widely used Easy installation Low economic cost	Sensitive to environmental noises
AE sensor	AE	High frequency response Roust to interferences	Hard to be analysed effectively
Hall sensor	Current/Power	Easy installation Low economic cost	Capture limited information

Even though the motor current/power can reflect the change in cutting force resulting from tool wear. Experimental results showed that in high-speed machining, the current was not sensitive enough to the cutting force because most of the energy was used to drive the spindle rotation [57]. In addition, the current or power measurements may be confounded by circuit noise and frictional interactions among mechanical components. The monitoring accuracy using motor current/power still needs to be improved.

Table 2.2 summarises the characteristics of the sensors discussed above. Note that other sensing technologies have also been reported in the literature, such as the sound [58], temperature [59], torque [60], and chip reflectance [61]. It can be found that a TCM system employing a single sensor more or less has some drawbacks. In the framework of multi-sensor TCM, the lack of information in one sensor signal might be compensated by another sensor. Hence, researchers have put more attention on designing the TCM system via multi-sensor configuration. This trend is also reflected by the increasing number of publications associated with the multi-sensor data-driven TCM strategy [8, 32, 62–64]. But the accuracy is not necessarily improved with more sensors because of the redundancy of similar signals. A more efficient and cost-effective approach involves deploying a suite of key sensors while prioritising the development of advanced signal processing algorithms.

2.3 Feature extraction technique

Before feature extraction, the raw signals need to be pre-processed by segmentation and filtering. Segmentation is employed to isolate data segments corresponding specifically to continuous cutting passes, rather than idle strokes. Furthermore, some analysis methods only focus on the entering of each cutting tooth into the workpiece. Another important pre-processing method is low-pass or band-pass filtering, by which the high-frequency noises should be removed. When the pre-processing has been completed, different data processing algorithms can be carried out to construct some representative indices. Ideally, these indices are expected to be highly associated with the tool condition and be robust enough to environmental interference. In this section, features in three typical domains will be introduced, namely the time-domain, frequency-domain, and time-frequency domain features.

2.3.1 Time-domain features

The sensor signal is always collected at a constant sampling rate. The most common time-domain features are descriptive statistics, such as mean, variance, RMS, etc. The time series analysis technique has also been used to capture the dynamics of the cutting process. Common models include autoregressive (AR), moving average (MA), and autoregressive and moving average (ARMA). Then the model coefficients are used as features. In TCM, the data features are extracted repeatedly every time a new data segmentation is available till the end of the tool life. One set of features should reflect the tool condition over a certain machining period. These features ideally exhibit a robust correlation with measurements of tool wear.

For simplification, the sensor signal is denoted as $x_n, n = 0, 1, \ldots, N-1$, where N is the number of samples. The formulas of common statistical and model-based features in the time domain are shown in Table 2.3.

In this table, mean, variance, skewness and kurtosis are the first to fourth-order statistical distribution moments of the data. Mean and variance are well known, which represent the average value of data points and the degree of fluctuation around this mean value, respectively. Skewness is the third-order moment normalised by the cube of standard deviation. It is a measure of the symmetry of the data distribution. Kurtosis quantifies the extent of the tails of a data distribution relative to a normal distribution. A data set with high kurtosis tends to have more outliers. The next feature, RMS reflects the signal energy information instead of the waveform. It is an effective indicator of the overall power content of the signal. But RMS is not sensitive to the transient pattern. Peak and range are linked to the maximum and

Table 2.3: Time-domain features and associated formula

Feature	Formula
Mean	$\bar{x} = \frac{1}{N} \sum_{n=0}^{N-1} x_n$
Variance	$\sigma^2 = \frac{1}{N-1} \sum_{n=0}^{N-1} (x_n - \bar{x})^2$
Skewness	$x_{ske} = \frac{1}{N-1} \sum_{n=0}^{N-1} (x_n - \bar{x})^3 / \sigma^3$
Kurtosis	$x_{kur} = \frac{1}{N-1} \sum_{n=0}^{N-1} (x_n - \bar{x})^4 / \sigma^4$
RMS	$x_{rms} = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} x_n^2}$
Peak	$x_{peak} = \max(x_n)$
Range	$x_{range} = \max(x_n) - \min(x_n)$
Crest Factor	$x_{cr} = x_{peak}/x_{rms}$
Count	$x_{count} = \text{Count}(x_n > Threshold)$
AR Coefficients	$x_n = c + \sum_{lag=1}^{p} \phi_{lag} x_{n-lag} + \varepsilon_n$, where p is the model order
MA Coefficients	$x_n = \mu + \varepsilon_n + \sum_{lag=1}^{q} \theta_{lag} \varepsilon_{n-lag}$, where q is the model order
ARMA Coefficients	$x_n = c + \varepsilon_n + \sum_{lag=1}^{p} \phi_{lag} x_{n-lag} + \sum_{lag=1}^{q} \theta_{lag} \varepsilon_{n-lag}$, where p is the order of AR term, q is the order of MA term

minimum amplitudes of the signal. Crest factor is defined as the ratio of the peak to the RMS as shown in Figure 2.3. It indicates how extreme the peak is in the waveform. Thus this feature can be used to detect instantaneous changes in the cutting process. Count is the number of samples exceeding a pre-set threshold.

In addition to statistics, some researchers have focused on the process dynamics by modelling the sensor data. Two types of model have been widely used, which are physical model and time-series model. Physical model refers to a mathematical model describing the physical relationship between the tool condition and other process variables. It directly provides the estimation of varying tool state at any time point [12–14]. However, due to the uncertainty of external factors, a deterministic model cannot sufficiently represent the machining process. The time-series modelling aims to build a stochastic model with discrete input/output data, expressing the system dynamics from a probabilistic point of view [65]. This method is totally data-driven. The underlying tool condition can be obtained via the model parameters or system features. For instance, Liang and Dornfeld [66] estimated an AR model from the AE signal and organised the model parameters as a feature vector. The analysis result suggested that the evolution of this vector was related to the change in tool wear. The statistics of the residuals, which are the difference between model predicted outputs and true values, can also reflect the disturbance of machining condition or workpiece machinability [42, 67]. Recently, a novel sensor data modelling and model frequency analysis-based method has been proposed for detecting tool anomalies [68]. The nonlinear relationship between input and output vibrations was modelled by a nonlinear autoregressive exogenous model (NARX), from which the nonlinear output frequency response functions (NOFRFs) were identified to represent process dynamics. The results proved that the model features had a better generalisation performance in comparison with signal statistics.

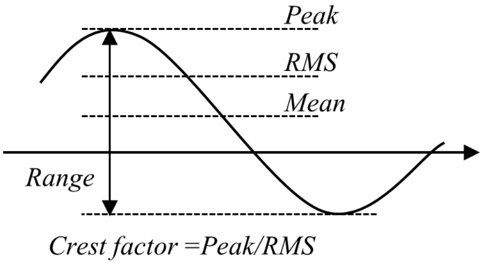


Figure 2.3: Diagram of several time-domain features.

2.3.2 Frequency-domain features

In the frequency domain, signals are depicted through the frequency spectrum, illustrating the amplitudes of distinct frequency components. As the periodicity of the cutting process induces periodic changes in the sensor signal, the component associated with the cutting tool condition can be easily extracted by checking the tooth passing frequency (TPF). The frequency analysis is based on the discrete-time Fourier transform (DTFT) and the discrete Fourier transform (DFT). The DTFT of a piece of signal x is written as

$$X(e^{jf}) = \sum_{n=0}^{N-1} x_n e^{-jfn} \tag{2.1}$$

where f denotes discrete frequency. DFT is the DTFT, eq. (2.1), evaluated at $f_k = 2\pi k/N, k = 0, 1, \ldots, N-1$, with a discrete frequency resolution of $2\pi/N$. As the DTFT/DFT results are complex values, the frequency-domain graph shows both the amplitude and phase spectra of the signal. For simplification, the amplitude of DFT $X(e^{j2\pi k/N})$ is written as $X_k, \quad k = 0, 1, \ldots, N-1$. The common frequency-domain features are shown in Table 2.4.

The formulas of most frequency-domain features are similar to those in the time domain. However, in the frequency domain, the calculation may not necessarily utilise all amplitudes across the whole frequency range. Instead, researchers can focus on the amplitudes over a certain frequency range or at a certain frequency and its harmonics, such as the tool

Table 2.4: Frequency-domain features and associated formula

Feature	Formula
Mean	$\bar{X} = \frac{1}{N}\sum_{k=0}^{N-1} X_k$
Variance	$\sigma_X^2 = \frac{1}{N-1}\sum_{k=0}^{N-1}\left(X_k - \bar{X}\right)^2$
Skewness	$X_{ske} = \frac{1}{N-1}\sum_{k=0}^{N-1}\left(X_k - \bar{X}\right)^3/\sigma_X^3$
Kurtosis	$X_{kur} = \frac{1}{N-1}\sum_{k=0}^{N-1}\left(X_k - \bar{X}\right)^4/\sigma_X^4$
Peak	$X_{peak} = \max\left(X_k\right)$
Peak Frequency	f_{peak}, the frequency where $X\left(f_{peak}\right) = X_{peak}$
Crest Factor	$X_{cr} = X_{peak}/\bar{X}$
Harmonics of TPF	$X_{tpf} = \sum_{k=1}^{K} X\left(f_k\right)$, where f_k denotes the k-th order harmonic of TPF

frequency and tooth passing frequency. Other frequency-domain analysis methods have been reported in the literature as well. As introduced by Kilundu et al. [69], effective features could be extracted from the singular spectrum of the vibration signal. Ai et al. [70] utilised the acoustic spectrum and linear predictive cepstrum coefficients of the sound signal to produce effective indices of cutting tool conditions.

2.3.3 Time-frequency-domain features

Note that a common assumption in the calculation of both time-domain and frequency-domain features is that the signal is stationary, which is not always true for the signals collected from a cutting process [39]. This shortcoming can be lessened by the time-frequency analysis techniques. Initially, the wavelet transform (WT) is proposed to obtain the time-frequency representation of a sensor signal. The basic concept is to decompose the signal into different scales. The highest scale corresponds to the low-frequency content or the general trend of the signal. With the decrease of the scale, the components at higher frequencies remain. The WT yields a time-frequency spectrum, enabling the extraction of amplitude values for each frequency band at any given time point. The popular WT algorithms include continuous WT (CWT), discrete WT (DWT), and wavelet packet transform (WPT).

In wavelet analysis, the signal is decomposed based on a function family $\psi_{b,a}(t)$. These functions are generated from a wavelet basis function $\psi(\cdot)$ through scaling and time-shifting operations. The relationship is described as

$$\psi_{b,a}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \tag{2.2}$$

where a defines the scale and b denotes the time shift.

Then, the CWT of the signal $x(t)$ is defined as the inner product of $x(t)$ and the wavelet basis function, i.e.,

$$CWT_x(b,a) = \frac{1}{\sqrt{a}}\int_{-\infty}^{+\infty} x(t)\psi\left(\frac{t-b}{a}\right)dt = \langle x(t), \psi_{b,a}(t)\rangle \tag{2.3}$$

In practice, the CWT algorithm is not computationally efficient. The DWT is proposed as a fast and efficient algorithm with $a = 2^j$, $b = k2^j$ [71]. In DWT, a two-scale relationship between the so-called mother wavelet and father wavelet is defined to calculate the wavelet coefficients directly,

$$DWT_x(j,k) = 2^{-j/2}\int_{-\infty}^{+\infty} x(t)\psi\left(2^{-j}t-k\right)dt \tag{2.4}$$

In CWT and DWT, there is a trade-off between the frequency resolution and temporal resolution. It means in high-frequency bands, the temporal resolution is fine while the frequency resolution is limited and vice versa. This limitation is resolved by the WPT algorithm which further decomposes the high-frequency components into more scales. Hence, the high-frequency information can be subdivided and analysed at a higher resolution.

In the TCM field, many researchers select the WT coefficients as representative features. Torabi et al. [72] extracted the energy of different scales from the CWT results of force and vibration signals and fed them into a fuzzy clustering system. The result showed that the CWT features recorded changes in the tool condition. Madhusudana et al. [73] found the DWT features of the sound signal performed better than common statistics in a face milling experiment. Similar findings have also been reported in the comparison between the WPT features and time-domain features of sound and vibration signals [74]. It should be pointed out that the performance of the WT features is determined by the wavelet basis function. An appropriate basis function should be similar to the sensor signal waveform as closely as possible. While it is difficult to choose such a basis function in real-time.

In recent years, another time-frequency analysis method named Hilbert-Huang Transform (HHT), which does not rely on the basis function, has been proposed. The HHT algorithm consists of two steps of empirical mode decomposition (EMD) and Hilbert transform [75]. The

EMD is used to decompose a signal $x(t)$ into N intrinsic mode functions (IMFs) $c_n(t)$ and a residual $r(t)$ in an adaptive way as $x(t) = \sum_{n=1}^{N} c_n(t) + r(t)$. Each IMF component corresponds to a specific frequency band. Instead of using any basis functions, the EMD is carried out by fitting the upper and lower envelope curves from the signal waveform, extracting the average of the two curves as an IMF, and decomposing the residual iteratively. In theory, this decomposition is applicable to any signal, including stationary and non-stationary signals.

After the decomposition, Hilbert transform is applied to each IMF component. It is defined as a convolution of $c_n(t)$ with the signal $h(t) = 1/\pi t$ as given by

$$H_n(t) = \frac{P}{\pi} \int_{-\infty}^{+\infty} \frac{c_n(\tau)}{t-\tau} d\tau \tag{2.5}$$

where P is the principal value of the integral. The Hilbert spectrum can be obtained according to the Hilbert transform result. The spectrum displays the time-frequency distribution of signal amplitudes. As this method is relatively new, the application in the TCM field is not reported widely, while researchers have already made some interesting attempts. Shi et al. [76] combined the EMD with independent component analysis to select representative IMFs from the sound signal. They observed that these components were associated with normal and broken tool conditions, which could be applied in tool breakage detection. Shen et al. [56] extracted tool wear-related features from the marginal spectrum of the HHT of spindle power signals in a drilling process. Figure 2.4 demonstrates typical time-frequency spectrum and IMF components. Nevertheless, the EMD algorithm faces problems of modal aliasing and computational inefficiency. More theoretical and practical studies are needed to improve its applicability in practice.

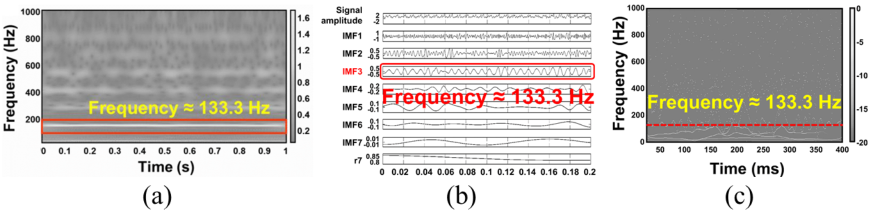


Figure 2.4: (a) Wavelet time-frequency spectrum; (b) IMF components; (c) Hibert-Huang time-frequency spectrum [56].

2.4 Dimension reduction of the extracted features

The use of multiple sensors and multi-domain features causes the problem of 'curse of dimensionality'. In order to reduce the computational burden and eliminate the risk of overfitting, the dimension reduction techniques are widely studied by researchers. Typically, two common approaches for feature reduction are employed, namely feature selection and feature fusion. The difference between these two approaches lies in whether the final features exist in the raw feature space or not. In this section, common feature reduction algorithms will be introduced.

2.4.1 Feature selection algorithms

Feature selection indicates selecting a subset of raw features, in which the dominant information is extracted and the redundant or unimportant features are discarded [77]. Dash and Liu [78] streamlined a typical feature selection process into four stages: creation of a candidate subset, evaluation of the subset via certain metrics, verification of stop criteria attainment, and performance assessment of the chosen subset. Based on the evaluation procedure, feature selection methods can be categorised into three main types, i.e., filter, wrapper, and embedded methods as shown in Figure 2.5.

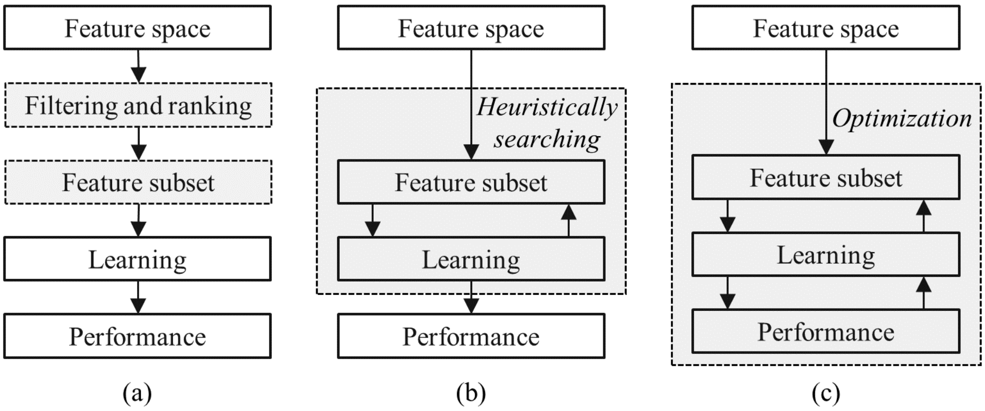


Figure 2.5: Framework of three typical feature selection methods: (a) filter method; (b) wrapper method; (c) embedded method.

The filter method operates by independently evaluating the performance of each feature, without taking the subsequent learning process into account [79]. Each feature is assigned a score based on a particular metric. These scores are then used to create a ranking of the features. Then the top features are selected based on a predefined threshold. Various metrics have been designed by researchers such as correlation relationship [80], feature similarity [81], mutual information [82], and relative contribution [83]. For example, in [84] and [19], the Fisher's discriminant ratio (FDR) was used to score the features' discriminability of cutting tool wear phases in advanced machining. In [62], Cho et al. extracted 135 features from 8 sensor signals and selected the most effective features using two methods based on a correlation metric and χ^2 statistics, respectively. The correlation-based feature selection method was proposed in [15], which selected an optimal subset of features that demonstrated minimal inter-correlation while maintaining a high correlation with the predicted class. Another method used the χ^2 test to evaluate the relationship between features and true classes. The features whose scores exceed a predetermined threshold were subsequently grouped together as an optimal subset. Zhou et al. [85, 86] proposed a dominant feature identification (DFI) scheme for feature selection. Suppose the singular value decomposition (SVD) of the feature matrix $\mathbf{F} = [f_1, f_2, ..., f_n] \in \Re^{m\times n}$ was given as $\mathbf{F} = \mathbf{U\Sigma V}^T$, where m was the length of samples and n was the number of candidate features. The first q rows of $\mathbf{\Sigma V}^T$ were extracted and denoted as $\mathbf{W} = \mathbf{\Sigma}_1\mathbf{V}_1^T = [w_1, w_2, ..., w_n] \in \Re^{q\times n}$. Then the columns of matrix $\mathbf{W}$ were grouped into p centroids using K-means clustering algorithm. Correspondingly, p column vectors in $\mathbf{W}$ having the shortest distance to each cluster centroid were determined. The indices of these vectors were expressed as $col[w_i], i = 1, ..., p$. In the raw feature matrix, the p feature vectors at the same positions were selected as the dominant features $[f_{col[w_1]}, f_{col[w_2]}, ..., f_{col[w_p]}]$. The analysis showed that the DFI-selected features could achieve the minimal total least-squares estimation error compared with other statistical methods such as principal component analysis (PCA) [87]. In the TCM field, the mutual information has also been used for the selection of tool wear-related features [88]. Although filter methods are computationally efficient, the selected features may not be optimal for the target learning algorithms due to the lack of guidance from a specific learning algorithm. This issue can be addressed in the wrapper and embedded methods.

The wrapper method evaluates the quality of features based on the predictive performance of a predefined learning algorithm. The feature subset that yields the best predictive performance is used as the selected ones [89]. Common search approaches include the sequential algorithm [25], Genetic Algorithm (GA) [16], Ant Colony Optimization (ACO) [90], and Grey Wolf Optimization (GWO) [91]. In [25], the authors extracted 151 time-domain and frequency-domain features from 8 sensor signals. A forward and backward sequential feature subset selection method was used to identify the optimal feature subset and finally only 13 features were selected. Zhou and Xue [16] extracted 175 features from 7 sensor signals. A kernel-based extreme learning machine (KELM) was employed as the regression model, in which the performance of the feature subsets was optimised by GA. During the search process, the optimization objective was defined as the root mean squared error between the predicted values of the KELM model and the actual values. Experimental results indicated that the GA-KELM-based features outperformed those selected by typical filter methods. Liao [90] extracted AR model coefficients and WT spectrum energy as an initial feature set. Then the ACO-based method and the sequential forward floating method were both used for feature selection. The objective function was defined as the performance of four different classifiers. The effectiveness of selected features was validated through grinding experiments. However, it should be noted that there are two specific limitations of the wrapper method that impede its practical application. Firstly, the selection process is quite time-consuming, as the machine learning model must be trained and tested for every possible feature subset and the size of the search space increases exponentially as the feature candidates increase [79]. The second limitation is related to the use of the model's accuracy as the objective function. This strategy may yield a feature subset that exhibits high accuracy on the training data but low generalisation capability.

The embedded method improves computing speed by integrating feature selection into the training process [92]. The contribution of each feature on the prediction results is assessed, and the most effective features are chosen. Various metrics have been proposed for ranking features, including the mutual information and its expansion, weights of a classifier, regularization parameters, and results of a decision tree model. As a relatively new method, the embedded method has not been widely applied in mechanical system condition monitoring. In [93], Yu proposed a logistic regression with penalization regularization (LRPR) method to select tool condition-related physical variables. The importance of each feature was identified by the penalization term. Then the top-ranked features were selected. The author utilised a logistic regression with manifold regularization (LRMR) model to assess the health state of the tool. The proposed method contained an adaptive learning scheme that reduced the number of parameters to be adjusted. Yang et al. [94] designed an embedded feature selection method for unsupervised outlier detection. The score of each feature was given based on the isolation forest and penalty value. Experimental results on three industrial datasets showed the proposed method could yield sensitive features, outperforming some common outlier indicators. Li et al. [95] proposed a feature selection method based on relevance vector machine with an approximated marginal likelihood. The posterior probability over coefficients and hyper-parameters of the prediction model was optimised. During this process, the feature selection was automatically performed. Case studies on two industrial datasets including the overflow particle size dataset and the mass of boiler stream dataset demonstrated the advantages of this method. As can be seen, the embedded method provides a trade-off between filter and wrapper methods, allowing to extract target-oriented features with less computational costs. In the future, it is expected to be a research hotspot in the TCM area.

2.4.2 Feature fusion algorithms

Feature fusion methods reduce the dimensionality of the feature space by combining multiple features from different domains and sensors into a single one. This method increases concentration of available information in raw feature space. In the literature, common fusion strategies include linear combination [96, 97], nonlinear mapping [17, 98], neural networks(NN)-based fusion [64, 99] and Bayesian inference-based fusion [100].

In linear feature fusion, each feature is assigned a weight that represents its relevance to the predicted targets. These weights are often determined based on the contribution of a feature to the decision making model or the correlationship between feature candidates. Typical algorithms include PCA and factor analysis (FA). PCA allows to identify a smaller number of features, the principal component scores, obtained through linear projection of the original features into a new space with reduced dimensionality, sufficient to describe the variance of the data. While FA is used for identifying latent variables that explain patterns of correlations in raw features. In [96], the author extracted 28 features from 7 sensor signals. According to the correlation with the tool wear, 8 features were firstly selected. Then the PCA was used to yield 2 principal features, representing the directions along which the data exhibited the largest variance.

Nonlinear feature fusion can be regarded as an expansion of linear methods, employing techniques like the kernel trick. The kernel trick involves a nonlinear transformation of the raw data, facilitating the computation of inner products between features in a high-dimensional space. Kong et al. [17] proposed a feature fusion method by combining kernel PCA (KPCA) with an integrated radial basis function. This method could achieve noise removal and dimension reduction at the same time. The final features generated a more compact and smoother confidence interval for Gaussian process model outputs, as shown in Figure 2.6. Wang et al. [98] introduced a feature fusion method based on probabilistic kernel FA (PKFA). The new model was able to handle multivariate features following a non-Gaussian distribution. Compared with conventional feature fusion methods such as PCA, KPCA, and KFA, the features derived from PKFA yielded more accurate tool wear prediction results.

Many researchers use machine learning models in feature fusion. Chen and Jen [99] applied different methods such as multiplication, summation, and division to fuse 9 independent variables. They introduced three new groups of fused indices into the original feature space. The performance of these generated features was evaluated within a NN-based learning model in terms of convergence speed and test error. A significant improvement in the monitoring accuracy was obtained when an appropriate fusion strategy was used. Ghosh et al.

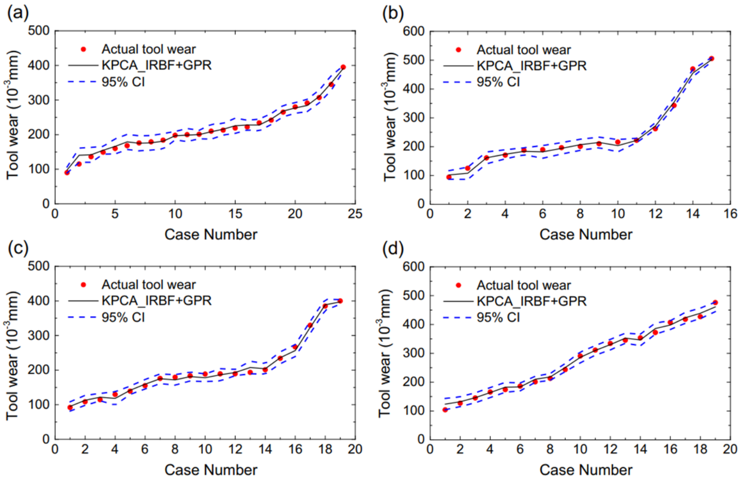


Figure 2.6: Predicted results of the GPR-based tool wear predictive model using the fused features of KPCA-IRBF in different tests [17].

[64] used the artificial neural network (ANN) as the feature fusion method. They first identified highly correlated signals, then fused the extracted features from different sensors using a fully connected back-propagation neural network (BPNN). The effectiveness of this method was validated through both laboratory and industrial experiments. In addition, Jaramillo et al. [100] developed a two-stage Bayesian inference data fusion scheme for monitoring the condition of distributed systems. They computed the posterior probability of a fault occurrence when certain features exceeded their respective thresholds while others did not, based on the prior probability and likelihood functions. This method enhanced the reliability of the health assessment of mechanical systems.

It should be noted that the methods mentioned above may yield different dimension-reduction results. For example, the features having high correlation relationship with the targets will be chosen if the Pearson's coefficient is adopted. While the features in different clusters will be selected if the DFI method is used. In the feature fusion strategy, the information of all features will be combined together. It is challenging to determine which strategy is better as all methods have been proved effective in specific cases. Therefore, the selection and validation of the algorithm should be conducted carefully, considering the characteristics of raw features as well as the unique requirements of specific applications.

2.5 Tool condition monitoring model

The decision-making system utilises the dominant feature subset as input to forecast the tool condition. Typically, the TCM system finds application in tasks such as tool fault diagnosis, tool breakage detection, tool wear monitoring, and prediction of remaining useful life. Within this system, a variety of ML models are integrated as decision-making tools, which can be broadly categorised into unsupervised and supervised learning approaches.

Unsupervised learning methods are commonly applied for tool breakage detection. The objective is to differentiate data representing abnormal states from those representing normal conditions without relying on sample labels. In machining processes, abnormal conditions typically indicate tool breakage or severe wear, where the tool completely loses its machining ability. The unsupervised learning model is designed to detect the occurrence of abnormal events rather than identifying specific types of faults.

In contrast, supervised learning methods offer more detailed insights into the tool condition due to the availability of a training dataset. In a classification model, each sample is assigned to a specific class. For instance, tool diagnosis outcomes may include various tool wear stages, types of wear, or even the locations where faults occur. Meanwhile, the objective of a regression model is more ambitious, aiming to generate continuous prediction values. For example, in a progression wear process, the regression model would estimate the current amount of tool wear or the remaining useful life of the tool.

2.5.1 Unsupervised model

The unsupervised learning model in TCM is typically an anomaly detection algorithm. In general, an index measuring the distance from the sample to the baseline is calculated and then compared with a predefined threshold. Such an index indicates how far one sample is away from the distribution of normal samples in the sense of statistics. If the index exceeds the threshold, the sample will be classified as an anomaly and the system will trigger an alarm.

Unsupervised learning-based TCM method has not been widely studied. Related research works have been reported in the 1990s and recent years [53]. Burke [101] utilised the unsupervised adaptive resonance network instead of BPNN to differentiate between sharp and worn cutting tools. The proposed scheme eliminated the need for extensive training data. Jammu et al. [102] also used an unsupervised NN for real-time tool breakage detection. The baseline condition was updated by a single category-based classifier in the machining process such that the true anomaly could be detected. However, the definition of tool breakage is referred to as severe failure, while in practice, minor tool failures are common and might also deteriorate the product. Hence, this method may not be suitable for practical applications.

Clustering analysis is another popular unsupervised algorithm. In [103], Li and Elbestawi proposed an automated TCM approach based on fuzzy logic and clustering algorithms. The tool condition was defined into three and five classes, respectively. In two cases, the proposed method achieved high classification accuracy.

McLeay [53] has made some interesting attempts to develop an unsupervised TCM system. For anomaly detection, he used the Mahalanobis distance as the monitoring statistic and determined the threshold using the Monte-Carlo algorithm. By assuming a Gaussian distribution for the normal samples, anomalies could be identified within a certain confidence interval. The experimental results showed that the definition of normal condition, which was pre-estimated based on empirical knowledge and not accessible during the machining process, significantly influenced the detection accuracy as shown in Figure 2.7. For tool fault diagnosis, he utilised the Gaussian mixture model (GMM) to identify the transition points between different tool conditions. Similar findings have also been reported for assessing the degradation rate of cutting tools used in packaging machines [24]. Moreover, unsupervised condition monitoring can also be conducted by fitting an analytical model [104] or building statistical process control charts, such as Hotelling's T^2 chart [105] and the cumulative sums chart [106].

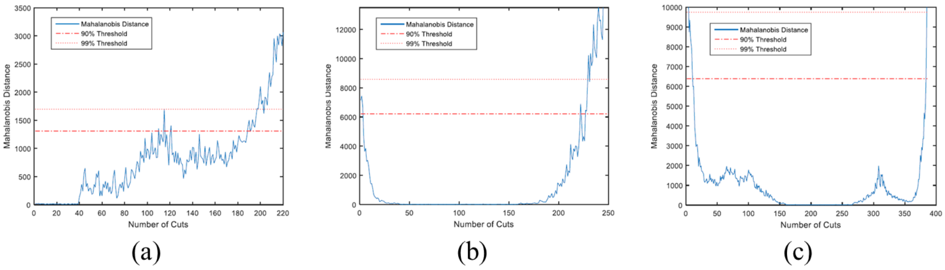


Figure 2.7: Anomaly detection results with different definitions of the baseline: (a) cut 1-39; (b) cut 40-159; (c) cut 160-266 [53].

2.5.2 Supervised Classification model

In TCM, there are two main type of classification problems: binary and multi-class. Binary classifiers can only classify data into two categories, while multi-class classifiers can classify data into more than two categories. Instead of estimating continuous and real-time changes in tool conditions, classification models allow operators to discern the overall state of the tool. This includes determining whether the tool is in good condition or failure, or identifying the specific wear stage the tool is currently experiencing. To build an accurate classification model, careful definition of the tool states and representative training data sets are necessary.

The challenge lies in the complex tool conditions under varying machining settings.

Classification models have been extensively used in TCM. Cho et al. [15, 62] compared the performance of various classification models in a multi-sensor fusion-based TCM scheme. The defined classes include breakage, chipping, and slight, medium, and severe wear. Time and frequency domain features were extracted from multi-sensor signals. Common models including the Support Vector Machine (SVM), Multilayer Perceptron NN (MLP), Radial Basis Function NN (RBF), and ensemble approach were used as classifiers. The classification accuracy of these models on the testing sets ranged from 84% to 97%, indicating that the tool conditions were correctly classified in most cases. Boutros and Liang [107] used the hidden Markov model (HMM) to detect and diagnose mechanical faults. The HMM assumes that the current system state is only related to the last state. Experimental results showed that this method could determine the severity and position of tool fault accurately. Hsieh et al. [108] adopted the BPNN to classify sharp and worn micro-tools. The class-mean scatter criterion was used to evaluate the sensitivity of the features to the changes in tool wear, which helped decide an optimal feature extraction procedure. Karandikar et al. [11] applied a naïve Bayesian classifier on both discrete and continuous TCM cases. For the discrete case, the tool condition was divided into new, mildly worn, and worn states. Corresponding force signal features were also divided into three classes. The posterior probability for the tool state given different features was updated using the Dirichlet-multinomial distribution conjugate pair, achieving a classification accuracy of about 85%.

The recent 10 years have witnessed a spur rapid progress in AI, especially in the realm of deep learning, due to the development of computing power, data, and optimisation techniques. Compared to traditional ML models, DL models often demonstrate superior learning and prediction capabilities when handling a high volume of sensor data. In ML models, feature engineering is crucial for accurate predictions. While DL models eliminate the need for feature extraction, as they implicitly conduct this process and directly establish the relationship between the sensor signal and predicted targets. The DL method has been successfully applied in condition monitoring [109, 110]. For example, Shi et al. [111] presented a fusion of multiple stacked sparse auto-encoder (SSAE) for TCM. This method comprised two structures that learned lower-level and higher-level features, respectively. The tool condition was grouped into four categories based on the flank wear width. The proposed model achieved an average testing accuracy of up to 96%, outperforming popular ML methods and standard DL methods. Serin et al. [40] highlighted the immense potential of DL methods in the TCM field, particularly in the context of industry 4.0. They demonstrated an example of utilising convolutional NN (CNN) for tool wear classification as shown in Figure 2.8.

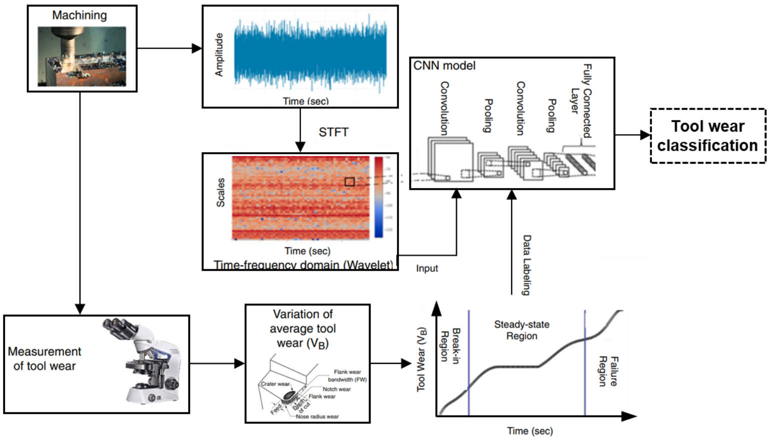


Figure 2.8: Example application of CNN in tool wear classification [40].

2.5.3 Regression model

The regression model is used to estimate the current tool state and predict the tool's RUL in the TCM field. The prediction accuracy is significantly influenced by the learned model parameters and structures. Ideally the training data set should contain sufficient and different samples corresponding to different tool conditions. However, the tool wear state is influenced by many factors such as cutting conditions, workpiece material machinability, the design of tools, and the use of coolant. Consequently, obtaining a comprehensive training set is still a challenge in tool wear prediction, owing to the high time-consume and economic cost.

Both ML and DL models have been used in the regression task. Shi and Gindy [112] developed a tool wear prediction model based on the least squares SVM (LS-SVM). Its effectiveness was quantified by the mean square error of the predicted results. But the performance in different tools and conditions was not tested. Li et al. [113] utilised a fuzzy NN in tool wear estimation. The integration of fuzzy rules expedited parameter learning and improved prediction accuracy. Despite outperforming BPNN and RBF, the definition and optimization of fuzzy membership items still required expert knowledge. In [11], the authors investigated the application of Bayesian prediction models in both discrete and continuous TCM scenarios. For the latter case, an ordinary Bayesian linear regression model was used for tool wear monitoring, providing both predicted tool wear and probability density. However, the prediction accuracy relies on the quality and representativeness of the training data, as well as the similarity between the distributions of the training and testing datasets. To overcome the difficulty of limited training data, Drouillet et al. [55] proposed a simple method to generate pseudo samples. These samples are functions of the power signal feature and machining time. The feature values and machining time were assumed to follow a predefined distribution. According to the distribution, they simulated uncertainty from different conditions and environmental noises by generating 1000 virtual curves, significantly enlarging the training set. Thus the RUL of the tool could be updated in real-time by finding the best-fitted curve, as shown in Figure 2.9. Kong et al. [17] applied the Gaussian process regression (GPR) model for the estimation of tool flank wear. The GPR model quantified the intensity of Gaussian noise, calculating both the predicted tool wear value and the confidence interval. Recently, a variety of DL models, such as meta-learning [113], deep transfer learning [114], and CNN-long short-term memory (CNN-LSTM) network [21], have been employed for tool wear estimation and RUL prediction.

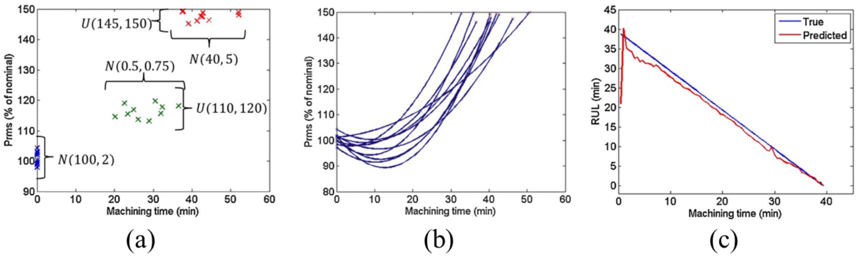


Figure 2.9: (a) Uncertainty distribution determined from the experimental data; (b) examples of generated curves; (c) prediction result of RUL. [55].

A benchmark dataset is crucial for evaluating the performance of different models as it provides a consistent basis for comparison. In the TCM field, an open dataset was provided in the 2010 Prognostics and Health Management (PHM) data challenge [115]. The experimental setup is shown in Figure 2.10. This dataset consisted of multi-sensor signals from each cutting pass. Six end mills were used in the experiment, with flank wear measured for three of them after each cutting pass. Various regression models, including support vector regression (SVR) [98], nonlinear function [116], kernel extreme learning machines (KELM) [16], LSTM [20], and GPR [117] have been developed using this dataset. However, a significant challenge arises when attempting to accurately predict tool wear of one cutter based on training data from other two cutters. In [98], the authors utilised PKFA algorithm to fuse the extracted features and trained a regression model based on SVR. This method showed a very accurate prediction performance. It indicates that the generalisation ability of regression models is influenced by the consistency between training and testing samples, and the size of training datasets. This benchmark dataset is collected from a simplified face milling process, which does not consider the variations of machining strategy, tool types, and workpiece material. Hence, the trained model cannot be readily used in other machining processes. The study of PHM for cutting tools should be practice-oriented, which means the trial cuts are designed according to production requirements and improve the generalisation capability of the model within a specific type of machining process. Recently, the development of transfer learning model provides a new way to transfer the knowledge obtained from one process to another similar one [114]. This technology may further improve the generalisation performance of TCM models. In this sense, the whole TCM community would benefit from more available datasets.

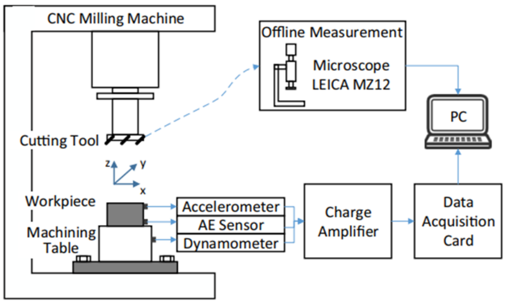


Figure 2.10: Illustration of the experimental setup for 2010 PHM data challenge dataset [20].

Table 2.5 compares the characteristics of three decision-making systems used in the PHM of cutting tools. All three types of models have been applied to achieve different monitoring targets. Unsupervised learning methods are useful for breakage detection and identifying fault-induced patterns in data, classification models are employed for evaluating tool health states or fault types, and regression techniques are utilized for predicting continuous variables related to tool wear or remaining useful life. It is worthwhile mentioning that a semi-supervised learning method that falls between supervised and unsupervised learning attracts attention from the TCM field as well [40]. The goal is to train a model on limited labelled samples, and improve the model performance by leveraging additional unlabelled data. This new learning paradigm utilises all available information more effectively. In practice, the specific objectives of the monitoring task and the nature of the dataset should be considered carefully before choosing which type of model to use. In the era of smart manufacturing, the design of an accurate and applicable TCM system will still be a research hotspot.

2.6 Summary

This chapter presents a comprehensive review of four essential steps of a data-driven TCM method, namely multi-sensor system, feature extraction, feature reduction and decision-making. To completely capture process information, researchers have been trying to integrate multiple sensors into the TCM system. Then a variety of features are extracted using time, frequency, and time-frequency-domain analysis methods. The subset of representative features is produced by the feature selection or feature fusion algorithm. Finally, the decision-making model is built to predict tool conditions. In addition, the properties of commercial TCM systems are also mentioned. Most of these systems only focus on the detection of tool

Table 2.5: Comparison of decision-making systems used in the PHM of cutting tools

Model	Unsupervised	Classification	Regression
Objective	Identify patterns or anomalies	Classify samples into different categories	Predict continuous values
Input data	Unlabelled	Labelled	Labelled
Output	Clusters, anomalies	Categorial	Continuous
Algorithm	K-means clustering, anomaly detection	SVM, MLP, SAE	SVR, GPR, LSTM
Application	Breakage detection [87], fault diagnosis [53]	Fault diagnosis [111], fault localisation [62]	Wear estimation [11], RUL prediction [116]

failure, such as tool breakage or severe wear. While the academia pays more attention to the RUL prediction of the tool, which is the basis of predictive maintenance.

According to the literature review, the following three challenges still exist in the current PHM technology for cutting tools.

1. The limited and imbalanced training data constrains the generalisation performance of the decision-making model.

2. The complicated machining processes produce nonstationary measurements and unexplained interference, for instance, both normal process degradation and faulty events might trigger alarms. The challenge lies in how to enhance the robustness of the monitoring system to normal changes and disturbances but remain its sensitivity to fault symptoms.

3. The use of multiple sensors and various signal features leads to the curse of dimensionality bringing about high computation cost and overfitting risk.

This project aims to deal with these challenges using the novel time-sensor domain data decomposition and analysis framework. Details of the proposed methods will be demonstrated in the following chapters.

3

Time and sensor domain data decomposition and analysis framework

3.1 Introduction

This chapter will introduce the basic idea and principle of the proposed time and sensor domain data decomposition and analysis framework. This novel idea is inspired by the concept of time-space decomposition developed in DPSs modelling [27, 28]. The research status of time-space decomposition and its applications are reviewed first. Then the concept of time-sensor decomposition is introduced. Furthermore, the specific methodology regarding the decomposition and analysis procedure is proposed, with a fault diagnosis model presented. The benefit of this novel framework is demonstrated through two case studies where the time and sensor domain components are directly applied to tool fault diagnosis in milling processes. The time and sensor domain features provide an overall high accuracy and good generalisation capability compared with raw signal feature-based methods. Practical issues related to the implementation of the proposed framework are discussed as well.

3.1.1 Time-space decomposition-based system modelling

As mentioned in Chapter 1, time-space coupling systems are widespread in science and engineering. The system dynamics is usually modelled by PDEs, which are difficult to solve directly. A common way to resolve such a large and complicated system is to decompose it into relatively small sub-systems and study the behaviour of each sub-system [118]. Time-space decomposition is a common idea for the analysis of time-space coupling systems. Suppose the system state at time t and position (x_1, x_2, x_3) is denoted as $\Phi(x_1, x_2, x_3, t)$, the system state can be decomposed as the linear combination of independent spatial basis functions $\varphi(x_1, x_2, x_3)$ with respect to the time coefficients $y(t)$ as

$$\Phi(x_1, x_2, x_3, t) = \sum_{i=1}^{N} y_i(t) \cdot \varphi_i(x_1, x_2, x_3) \approx \sum_{i=1}^{K} y_i(t) \cdot \varphi_i(x_1, x_2, x_3) \tag{3.1}$$

where N is the number of all sub-systems and K is the number dominant basis functions, from which most of the system dynamics can be recovered even if $K \ll N$.

Traditionally, the spatial basis functions (SBFs) can be selected as common orthonormal functions, including Legendre polynomials, Jacobi polynomials, and Fourier series, which rely on theoretical analysis of process dynamics. The Karhunen-Loéve decomposition algorithm avoids this limitation, determining empirical SBFs from the system response directly. Due to that, the KLD-base system modelling approach is also referred to as data-driven modelling, which has been commonly used to realise the time-space decomposition. Similar to SVD and PCA, KLD algorithm simplifies the system model based on the covariance structure of system variables or measurements. The difference among them is that, SVD is a general algorithm for the eigenvalue decomposition of both square and non-square matrices [119]; while PCA mainly focuses on reducing the interrelated random variables to a subspace of principal components [120–122]. As Lu et al. [123] pointed out, the SVD-based model reduction can be regarded as a more efficient algorithm. Both SVD and PCA methods can only handle discrete-time processes. While the establishment of KLD extends the application scenarios to continuous-time processes [124, 125]. The decomposition results of these three methods are equivalent for discrete cases.

Various applications of time-space decomposition in complex system modelling have been reported, covering scientific and engineering fields. The aim is to reduce model complexity and improve the efficiency of numerical solution [126]. In the analysis of fluid dynamics, the decomposition method can identify the coherent structures in the turbulence flow from two-point spatial correlation functions [127]. In the modelling of industrial thermal process, this method is applied to determine dominant spatial and temporal BFs [27, 128, 129]. In structure health monitoring, Feeny and Kappagantu [130] extracted proper orthogonal modes from historical displacement measurements collected by distributed sensors. Azeez and Vakakis [131] studied the lower-dimensional modes of vibroimpacting systems based on the shapes of dominant modes. The time-space decomposition can also be applied to problems in extreme scales from the dynamics of micro-electromechanical systems (MEMS) [132, 133] to the analysis of remote sensing observations of our planet regarding climate changes, temperature, and carbon intensity [134–136]. Moreover, by decomposing the distributed ECG signals, researchers can learn about the latent temporal and spatial components related to people's health [137, 138].

However, in the area of mechanical system condition monitoring, the use of the time-space decomposition approach is rarely reported. To the best knowledge of the author, the KLD approach was initially employed by Tumer for monitoring the condition of manufacturing processes [139–141]. The input signal was the surface profile measurement. In total, $M = 60$ input profile measurements were collected, with $N = 1000$ points each as shown in Figure 3.1 (a). The different measurements were collected in the direction perpendicular to the longitudinal. Figure 3.1 (b) shows some input profiles. All the measurements were organised as a 60×100 matrix. After the KL decomposition, the main three modes and corresponding coefficient vectors were extracted as shown in Figure 3.1 (c). The three modes consist of a linear trend, a low-frequency sinusoidal pattern, and a high-frequency sinusoidal pattern (twice the frequency of the first one). By analysing the characteristics of different modes, the machining condition can be monitored. For example, the first mode has a linear slope, reflecting misalignments of the worktable. The corresponding coefficient vector 1 indicates the change in slope and severity of this fundamental pattern along the transverse direction. Meanwhile, the second mode corresponds to the main periodic pattern generated on milled surfaces due to the feed marks during milling. Any changes in this component, or the relative severity of its magnitude, can be monitored and detected by the coefficient vector 2. Additionally, this KLD-based analysis was successfully applied to parts produced by selective laser sintering.

Tumer and Huff have also analysed the vibration signals of helicopter gearboxes using PCA [142, 143]. The signals were tri-axial vibration measurements of the gearbox housing. According to the experimental design, each flight consisted of 22 manoeuvres and was completed by two pilots. During one flight, 22 segments of data sets were generated and analysed separately. The PCA was performed to obtain the dominant vibration series and corresponding PCs from each data set, as displayed in Figure 3.2 (a) and (b). The coefficient vector was assumed to be constant unless the vibration signatures changed significantly. The authors analysed the change in the direction of this vector by the angle θ as shown in Figure 3.2

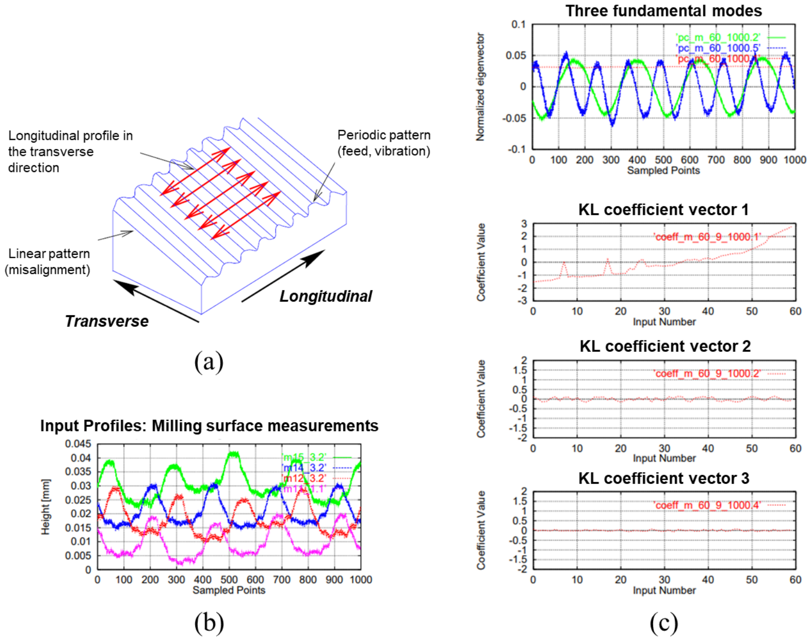


Figure 3.1: (a) Diagram of the milled workpiece and the measurement of profile; (b) Input profiles: milling surface measurements; (c) KL decomposition results: three fundamental modes and coefficient vectors [140].

(b). Figure 3.2 (c)-(f) show the changes of θ against the flight manoeuvres. It can be found that the same sequence of manoeuvres exhibits a similar pattern of change in the θ angle, regardless of the pilot executing them. The study involved the decomposition of multi-sensor data and confirmed that both the modes and coefficients were related to changes in the underlying process. But the theoretical foundation and its potential applications in the TCM field have not been established so far. The current research aims to fill this gap.

3.1.2 Time and sensor domain data decomposition

Even though time-space decomposition has been widely applied in DPSs modelling, the decomposition of multi-sensor data and its potential applications in process monitoring have not been established yet. This motivates the basic idea of the present study to apply the time-space decomposition to the analysis of multi-sensor data for industrial process monitoring.

In the analysis of DPSs, the magnitudes of a physical variable at different locations are measured by multiple sensors. By choosing dominant basis functions, the multi-sensor signals can be represented using several time coefficients and spatial basis functions. The time coefficients represent the change in system dynamics with time while the spatial basis functions are only dependent on spatial locations and described by data vectors. By contrast,

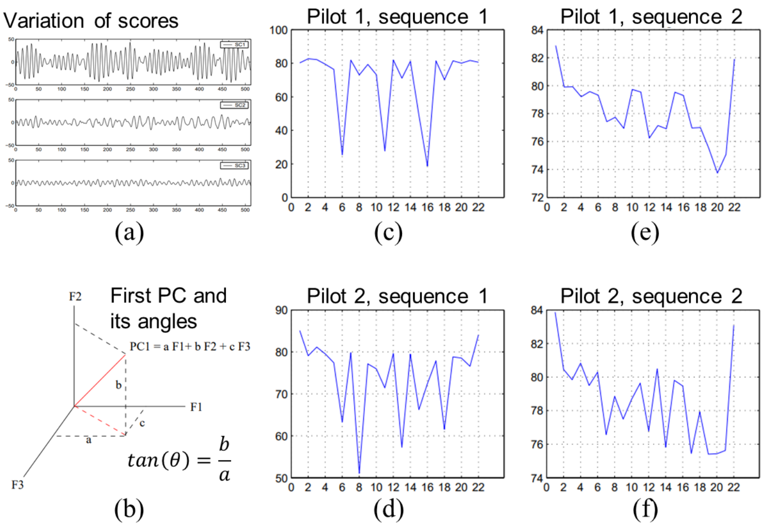


Figure 3.2: PCA results of tri-axial vibration data: (a) three time-domain series; (b) 3-dimensional coefficient vector; Changes of the θ angle of the coefficient vector vs. flight manoeuvres sequence: (c) pilot 1, sequence 1; (d) pilot 2, sequence 1; (e) pilot 1, sequence 2; (d) pilot 2, sequence 2 [142].

the data in the process monitoring field are usually collected from heterogeneous sensors. These sensors are not necessarily located in specific locations as the spatial association is not of our main interest. Mathematically, the decomposition of heterogeneous signals is the same as in the case of homogeneous signals. The generated time coefficients, which are named time domain components (TDCs) in the present study, still represent the temporal properties of the process. While the basis functions no longer represent different spatial location-related information but the information related to different sensors. This domain is, therefore, referred to as the sensor domain and the basis functions are called sensor domain components (SDCs) in the present study. Consequently, the new decomposition is referred to as time-sensor decomposition (TSD). There are two obvious benefits with the TSD methods. Firstly, the high-dimensional data can be compressed to low-dimensional space such that the analysis complexity is reduced. On the other side, the exploitation of the information dispersed among different sensors is expected to improve the monitoring accuracy.

Figure 3.3 shows a comparison of process monitoring approaches based on the conventional multi-sensor data analysis and the proposed time-sensor decomposition strategy. What both techniques have in common is to extract useful features from sensor signals and then feed these features into a decision-making model to infer the process condition. In Figure 3.3 (a), suppose 10 features are extracted from each sensor signal and 10 sensors are used in total, the number of features reaches 100. On contrary, the proposed approach starts with decomposing multi-sensor signals into a few pairs of TDCs and SDCs, and then extracts features from TDCs only, as shown in Figure 3.3 (b). Again 10 features are obtained from TDC1, and the SDC1 is a 10-dimensional vector. Thus only 20 features are used here, which means the number of features is reduced by 80% using the time-sensor decomposition strategy.

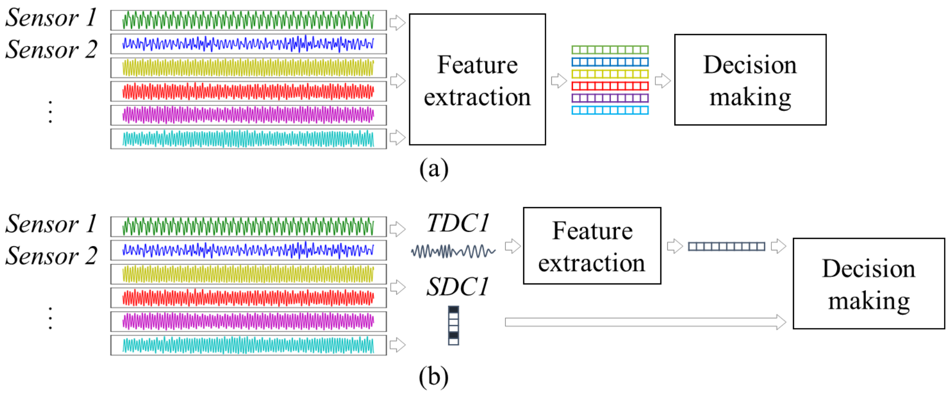


Figure 3.3: Illustration of multi-sensor data analysis based monitoring techniques: (a) conventional method; (b) the proposed time-sensor decomposition method.

The success of a decomposition approach relies on the fact that most of the raw information can be represented by a few pairs of TDCs and SDCs. By exploring this advantage, the condition-sensitive features can be extracted from the first few pairs of TDCs and SDCs and used to design an effective monitoring system. As mentioned above, TDCs record the common information contained in raw signals while SDCs reflect interaction information between involved sensors. Therefore, in condition monitoring of nonstationary processes, the TDCs and SDCs of the multi-sensor signals can be utilised together to reveal the changes in process states and provide a useful extra degree of freedom.

It should be noted that the condition monitoring is usually conducted on a segmentation of data. This implicitly indicates the process condition remains unchanging during the data collection period. Compared to the monitoring strategy on separate data samples, this method saves a huge amount of computational costs. However, the increase of monitoring interval will inevitably bring about delayed detection. In this case, domain knowledge is necessary to evaluate the requirements for computational efficiency and timely detection.

3.2 The proposed methodology

This section introduces the details of the proposed time-sensor decomposition algorithm. Then a time series analysis technique is applied to extract features from the TDCs. Finally, the time and sensor domain features are utilised to build a fault diagnosis model, which is also known as a classifier, aiming to detect whether the current process state is normal or faulty. Details of this approach are described as follows.

3.2.1 KLD-based time-sensor decomposition

The time-sensor decomposition is based on the discrete KLD algorithm, which can decompose multivariate data into linear combination of basis vectors [29]. Firstly, data pre-processing operation comprising low-pass filtering and resampling is applied to eliminate high-frequency noise and ensure uniformity in the number of data points across all channels. Then the multi-sensor data $\boldsymbol{x}_j \in \mathbb{R}^m,\ j = 1, \ldots, n$ constitute a matrix $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]$, where m is the number of sampling points and n is the number of variables, as demonstrated in Figure 3.4. The next step is to standardise each element by

$$x'_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j},\ i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n \tag{3.2}$$

where $\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_{ij}$ and $\sigma_j = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}\left(x_{ij} - \mu_j\right)^2}$ denote the mean and standard deviation of the j-th measurement $\boldsymbol{x}_j$, respectively. Then the covariance matrix of the normalised data matrix $\mathbf{X'}$ consisting of x'_{ij} is calculated by

$$\mathbf{C} = \frac{1}{m-1}\mathbf{X'}^T\mathbf{X'} \in \mathbb{R}^{n\times n} \tag{3.3}$$

As $\mathbf{C}$ is a positive definite matrix, it has an eigendecomposition

$$\mathbf{C} = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^T \tag{3.4}$$

where $\mathbf{\Lambda}= \mathrm{diag}(\lambda_1, \lambda_2,\ \ldots, \lambda_n)$ is a diagonal matrix containing the eigenvalues in a descending order, i.e., $\lambda_1 \geq \lambda_2 \geq\ \cdots \geq \lambda_n > 0$. The matrix $\mathbf{\Phi}= [\boldsymbol{\varphi}_1,\ \boldsymbol{\varphi}_2,\ \ldots,\ \boldsymbol{\varphi}_n]$ consists of normalised eigenvectors of $\mathbf{C}$. These vectors are defined as SDCs. The TDCs can be calculated by

$$\mathbf{Y} = \mathbf{X'}\mathbf{\Phi} = [\boldsymbol{y}_1,\ \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n] \in \mathbb{R}^{m\times n} \tag{3.5}$$

Note that $\mathbf{\Phi}$ is a unitary matrix. The decomposition of $\mathbf{X'}$ holds as follows

$$\mathbf{X'} = \mathbf{X'}\mathbf{\Phi}\mathbf{\Phi}^T = \mathbf{Y}\mathbf{\Phi}^T = \sum_{i=1}^{n} \boldsymbol{y}_i\boldsymbol{\varphi}_i^T \approx \sum_{i=1}^{n'} \boldsymbol{y}_i\boldsymbol{\varphi}_i^T \tag{3.6}$$

where n' denotes the number of dominant TDCs and SDCs. n' can be determined according to the magnitude of the eigenvalue $\lambda_i,\ i = 1, 2, \ldots, n$. As the data matrix has been normalised, $\lambda_1 + \lambda_2 + \cdots + \lambda_n = n$ holds naturally. The cumulative percentage of variance (CPV) of the

n' components is defined as the percentage of the total variances explained by the first n' eigenvalues, i.e., $CPV(n') = \sum_{i=1}^{n'} \lambda_i / n \times 100\%$. Then n' is selected when $CPV(n')$ exceeds a pre-defined threshold. In this research, for simplicity, we can select $n' = 1$, i.e., use the first pair of TDC/SDC corresponding to the largest eigenvalue as the dominant component.

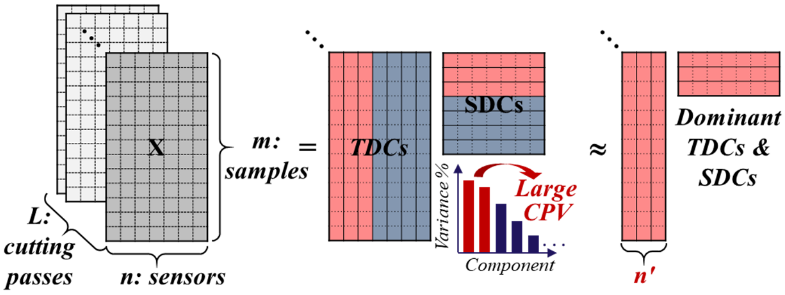


Figure 3.4: Illustration of the time-sensor domain decomposition of multi-sensor signals. The indication of the colours: red: dominant components; blue: insignificant components.

The meaning of TDC/SDC can be explained from the perspective of SVD as well. The normalised real matrix $\mathbf{X'}$ has a factorisation of the form $\mathbf{X'} = \mathbf{U\Sigma V}^T$, where $\mathbf{U}$ is an $m \times m$ real unitary matrix, $\mathbf{\Sigma}$ is an $m \times n$ rectangular diagonal matrix with non-negative real numbers on the diagonal, and $\mathbf{V}$ is an $n \times n$ real unitary matrix. The diagonal entries $\sigma_i = \Sigma_{ii}$ are known as the singular values of $\mathbf{X'}$, which is also sorted in descending order. Because the data are collected from different sensors and the number of samples is large enough compared to the number of variables, the rank of $\mathbf{X'}$ is n, which is also the number of non-zero singular values. Since $\mathbf{U}$ and $\mathbf{V}$ are unitary, their columns yield two orthonormal basis of $\mathbb{R}^m$ and $\mathbb{R}^n$ spaces, respectively. According to SVD, the matrix $\mathbf{X'}$ can be decomposed as,

$$\mathbf{X'} = \sum_{i=1}^{n} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T \tag{3.7}$$

Because the eigendecomposition of $\mathbf{X'}^T\mathbf{X'}$ can be re-written as $\mathbf{V}(\mathbf{\Sigma}^T\mathbf{\Sigma})\mathbf{V}^T$, the matrix $\mathbf{V}$ is exactly $\mathbf{\Phi}$ in 3.4. The TDC $\boldsymbol{y}_i$ defining as $\mathbf{X'}\boldsymbol{v}_i$ in 3.5, is also equal to $\sigma_i \boldsymbol{u}_i$. Thus, it is clear that SDCs are basis vectors of the row space of $\mathbf{X'}$, and TDCs are stretched basis vectors of the column space of $\mathbf{X'}$. In this sense, the data matrix is described in two separate spaces, and the dominant information is concentrated into the first several number of components.

3.2.2 Time domain data analysis

In principle, the feature extraction methods mentioned in Chapter 2 can be applied to analyse the TDCs. This section introduces a time series analysis-based approach, aiming to capture the process dynamics over a characteristic frequency range associated with the process of concern. Specifically, the power spectral density (PSD) of the TDCs are computed through building AR models. Then, the amplitudes of PSD over the characteristic frequency range are used as representative features. For this purpose, first of all, Seasonal and Trend decomposition using Loess (STL)-based algorithm is used to decompose the TDCs into trend (*T*), seasonal (*S*) and residual (*R*) components [144] as

$$y_i\left(t\right) = y_i^T\left(t\right) + y_i^S\left(t\right) + y_i^R\left(t\right),\ i = 1, 2, \ldots, n' \tag{3.8}$$

Through adjusting the parameters, the process-related information will be kept only in the seasonal or residual component. What is more, the obtained sub-component satisfies the wide-sense stationary assumption in AR modelling. For the residual component of the *i*-th TDC, an AR(*p*) model can be expressed by

$$y_i^R\left(t\right) = c_i + \phi_{i,1} y_i^R\left(t-1\right) + \cdots + \phi_{i,p} y_i^R\left(t-p\right) + \varepsilon_i\left(t\right),\ i = 1, 2, \ldots, n' \tag{3.9}$$

where *p* represents the model order, $\phi_{i,1}, \ldots, \phi_{i,p}$ are model parameters, c_i is a constant, and $\varepsilon_i(t)$ is the prediction error with zero mean and variance σ_ε^2. The model parameters can be determined using Yule-Walker equations [145].

From the AR model (3.9), a single-sided PSD of $y_i^R\left(t\right)$ can be obtained as

$$S_i\left(f\right) = \frac{2\sigma_\varepsilon^2 / f_s}{\left|1 - \phi_{i,1} e^{-j2\pi f/f_s} - \cdots - \phi_{i,p} e^{-j2\pi p f/f_s}\right|^2}, 0 \le f \le f_s/2,\ i = 1, 2, \ldots, n' \tag{3.10}$$

where f_s represents the sampling frequency.

It is well known that an AR(*p*) model has *p*/2 peaks in the frequency spectrum. Therefore, the model order is selected as a value between twice the number of $y_i^R\left(t\right)$'s spectrum peaks to be analysed and a reasonably large value that is related to the sampling rate and signal period. Then a cross-validation approach is applied to determine the optimal order that can produce the smallest multi-step-ahead prediction error.

3.2.3 Fault diagnosis model building

From the multi-sensor data, two types of features are extracted under the proposed TSD framework, which are dominant TDC features $\boldsymbol{f}_i = \left[S_i\left(f_1\right) \ldots\ S_i\left(kf_1\right)\right]^T$ with f_1 denoting the characteristic frequency, k being the number of harmonics of concern and dominant SDCs $\boldsymbol{\varphi}_i$, $i = 1, \ldots, n'$. Concatenating the n' dominant feature vectors produces a kn'-dimensional time domain vector $f_{time} = [f_1^T \dots f_{n'}^T]$ and an nn'-dimensional sensor domain vector $f_{sensor} = [\varphi_1^T \dots \varphi_{n'}^T]$.

All the available data are collected under normal and faulty conditions. Each sample consists of f_{time} and f_{sensor}. The corresponding label is set to 1 or -1 according to the process state. These datasets and labels are split into training and testing sets. During the data splitting, a stratified random sampling algorithm is adopted to ensure the data distribution in the subsets is the same as that in raw data set. Then a classifier can be built as follows,

(i) In the training stage, the hyper-parameters with the classifier are determined using a grid search approach with K-fold cross-validation (CV) strategy based on the time and sensor domain features and corresponding labels in the training set. After the CV procedure, the largest mean area under the curve (AUC) is recorded. Then the selected hyper-parameters are fixed and a classifier is trained using all the samples in the training dataset.

(ii) In the testing stage, the diagnosis performance of the trained model is tested using the time and sensor domain features in the testing dataset. Some metrics quantifying the model's classification ability are computed and evaluated.

The schematic flowchart of the proposed tool fault diagnosis approach is shown in Figure 3.5. Note that the presented time domain feature extraction is not unique. Other effective features can also be used under this framework.

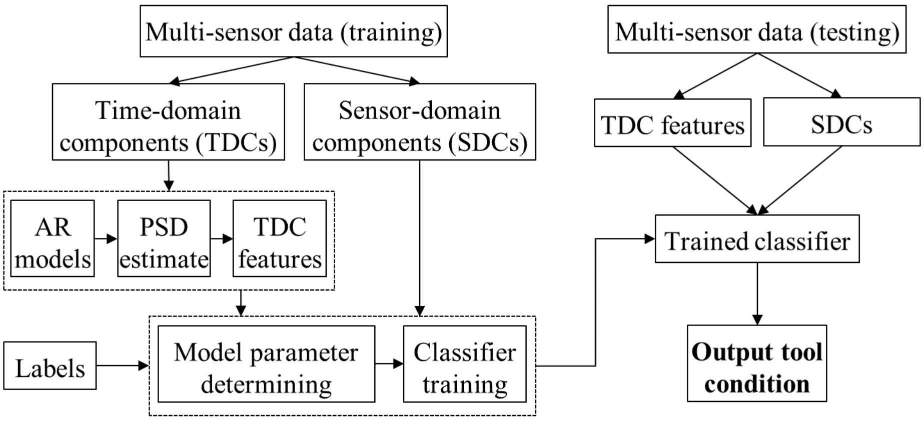

Multi-sensor data (training)
Time-domain components (TDCs)
Sensor-domain components (SDCs)
AR models
PSD estimate
TDC features
Labels
Model parameter determining
Classifier training
Multi-sensor data (testing)
TDC features
SDCs
Trained classifier
Output tool condition

Figure 3.5: Flowchart of the proposed TSD-based tool fault diagnosis approach.

3.3 Applications to fault diagnosis of cutting tools

The performance of the time and sensor domain features in process fault diagnosis is demonstrated in two milling experiments. The objective is to detect whether the tool condition is faulty or not. The state-of-the-art of the data-driven TCM system has been reviewed in Chapter 2. Despite the advantages such as easy implementation and the support to online monitoring, multi-sensor data-driven TCM approaches still face a big challenge in practice. That is, the use of multiple sensors and multi-domain features results in the so-called 'curse of dimensionality', which not only increases the computation burden but also reduces the monitoring accuracy.

To address this issue, the feature reduction methods such as feature selection and feature fusion [85, 87, 96] have been studied as shown in Chapter 2. As there is no additional requirement on the source signals, this approach is preferred in TCM. Besides, another two strategies have also been proposed to handle information redundancy. The first one is data-layer fusion method which can eliminate the measurement error and reduce data transmission burden [105]. This method is rarely used in tool fault diagnosis because of unclear relationship between the sensors. The second one is decision-layer fusion method that makes the final decision based on conclusions obtained from each sensor signal independently. The accuracy of this strategy relies on the design of the inference model such as majority voting and Dempster-Shafer theory [146]. It is widely acknowledged that the higher level the multi-sensor data is processed, the more information loss is caused [26]. For example, feature-layer fusion overlooks the correlation relationship existing among different sensors, while decision-layer fusion ignores weak evidence provided by each sensor. Except that, the influence of measurement error could be amplified in higher level approaches, which limits the generalisation capability of the trained model.

The TSD-based fault diagnosis approach can address these challenges. As the multi-sensor signals are decomposed into TDCs and SDCs directly, this framework can be deemed as an example of the data-layer strategy. It not only addresses the issue of high-dimensional features, but also reduces the influence of measurement error. The present study makes a preliminary attempt to apply TSD to tool fault diagnosis, showing an overall high accuracy and good generalisation capability.

3.3.1 Multi-sensor signal acquisition

Before introducing the experimental setup, the multi-sensor system used during the whole project will be described here. As mentioned in Chapter 2, the spindle current, vibration, force, and AE signal are the most popular signals in tool condition monitoring. Spindle current or power signal has been extensively investigated in recent years. Its value reflects the change in spindle load. Vibration signals have been utilised for decades in mechanical fault detection. The force sensor directly measures cutting force. Both vibration and force signals show a strong correlation with tool conditions. The AE sensor is used to detect frequencies over an ultrasonic range (several tens of kHz to several MHz), which result from crack damages.

In this study, the sound, vibration, and force signals are collected during the experiment. The sensor types and specifications are listed in Table 3.1. These sensors are mounted in positions as close to the machining area as possible. The accelerometers are mounted beside the workpiece and at the flange face of the spindle. The dynamometer is put beneath the workpiece. The CompactDAQ systems from National Instruments are used to collect data such that all measurements are synchronised. The sampling rates for all sensors are the same.

Table 3.1: List of sensors and specifications

Sensor	Model	Sensitivity	Frequency range /Hz	Channels
Free field microphone	PCB, HT378B02	50 mV/Pa	3.5-20000 (±2 dB)	1
Tri-axial accelerometer	PCB, 604B31	100 mV/g	0.5-5000	3
Tri-axial accelerometer	PCB, 356A02	10 mV/g	1-5000	3
Tri-axial dynamometer	Kistler, 9255C	x,y: -7.9 z:-3.9 pC/N	x,y: 0-2200 z: 0-3300	3

3.3.2 Case 1: milling process with straight-line path

Experimental setup

This experiment was conducted at Advanced Manufacturing Research Centre (AMRC), the University of Sheffield. The aluminium block-shaped workpieces were machined in a DMG MORI DMU 40 eVo machine centre. The cutting tool was a Sandvik CoroMill square shoulder end mill (316-25SM345-25000A H10F) with 25 mm diameter and 3 flutes. As shown in Figure 3.6, the cutter moves horizontally along the edge of the workpiece from left to right and then steps over to conduct next cutting. For one workpiece, the milling was repeated for 35 layers with 6 passes per layer. Two workpieces (denoted as W1 and W2) were firstly machined using brand new cutting tools, which were referred to as baseline (BL). Then another two workpieces (W3 and W4) were milled using worn cutting tools. This machining was referred to as toolwear (TW). The machining parameters were set as follows: the spindle speed was 6150 rpm, feedrate was 6260 mm/min, radial and axial depth of cut were 20 mm and 2 mm, respectively. As can be seen from Figure 3.6, two tri-axial accelerometers are mounted on the spindle and below the workpiece, respectively. The model of the sensor on the spindle is PCB 356A02. The sensor below the workpiece is PCB 604B31. Their specifications are shown in Table 3.1. The sampling rate was set to 51200 Hz. In total, six channels of vibration data were used in the analysis.

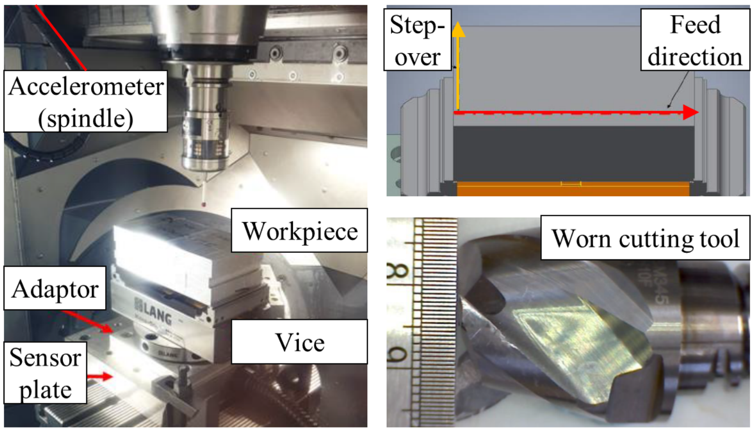


Figure 3.6: Experimental setup and the image of a worn end mill.

Multi-sensor data decomposition and analysis

In metal machining, the tooth passing frequency (TPF) describes the number of cutting edge contacts with the workpiece per unit time. The information over TPF is directly related to the tool condition. Considering the TPF in this experiment is 307.5 Hz, a low-pass filter is applied over the frequency range up to 1000 Hz then all signals are downsampled at 2460 Hz. After pre-processing, the multi-sensor data is decomposed into time and sensor domain by KLD algorithm. Table 3.2 shows the eigenvalues of the covariance matrix of signals captured during the first cutting pass under BL condition. As the first component takes up 50.6% of the total variance of raw signals, TDC1 and SDC1 are selected as the dominant component.

The next step is to extract features from TDC1, which aims to grasp cutting process-related

Table 3.2: Eigenvalues of the signals' covariance matrix, cut #1, BL

i	1	2	3	4	5	6
Eigenvalues	3.03	1.11	0.91	0.58	0.24	0.13
CPV (%)	50.6	18.4	15.2	9.7	4.0	2.1

dynamics using autoregressive models. First of all, the STL algorithm is applied to separate trend, TPF-related seasonal, and residual sub-components from TDC1. The obtained seasonal component only contains information at TPF and its harmonics. In this study, an AR(8) model is appropriate to fit TDC1's seasonal sub-component under BL condition and an AR(6) model is chosen for TW condition. The multi-step-ahead prediction results of two AR models built based on the data in the first cutting pass under BL and TW conditions are shown in Figure 3.7 (a) and (b). The identified model can accurately predict actual data with the root mean square error (RMSE) of 0.106 and 0.176. This indicates that the model can describe milling process dynamics accurately.

Figure 3.7 (c) presents the single-side PSD estimated using the aforementioned AR model parameters corresponding to BL and TW conditions. Both spectra show three peaks at 307.5 Hz, 615 Hz and 922.5 Hz but different amplitudes, which should result from different tool status. Here the spectral amplitudes at TPF and its second and third harmonics constitute the TDC1 feature vector.

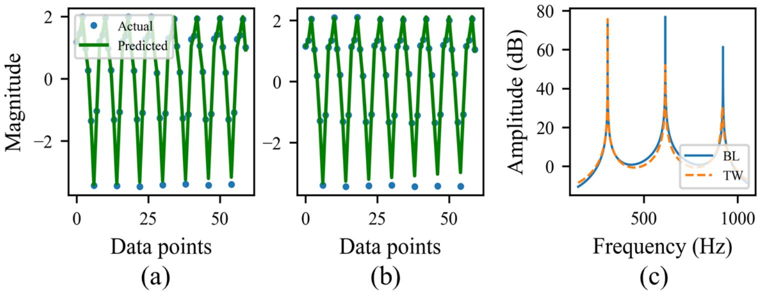


Figure 3.7: AR(p) model predicted outputs vs. actual values in cut #1: (a) under BL condition, p=8, RMSE=0.106; (b) under TW condition, p=6, RMSE=0.176. (c) the PSDs of TDC1's seasonal components in cut #1, under BL and TW conditions.

As another product of TSD, SDC1 is a 6-dimensional vector in this case. Figure 3.8 shows the SDC1s of all cutting passes under BL and TW conditions in the form of heatmap. In theory, the elements in the SDC vector reflect the coefficients of linear combination of raw signals to constitute the corresponding TDC. The changes in tool condition will lead to the changes in the correlation structure of multi-sensor signals, and thus the pattern of SDCs. From the heatmap, we can find that SDC1s show two distinct patterns. The main difference comes from the fourth element corresponding to the X-axial vibration on the spindle, which is around 0 in BL group and increases to 0.4 in TW group. Besides, the sixth element also shows slight difference in the two groups. This observation indicates that the SDC1s are also effective features in tool fault diagnosis.

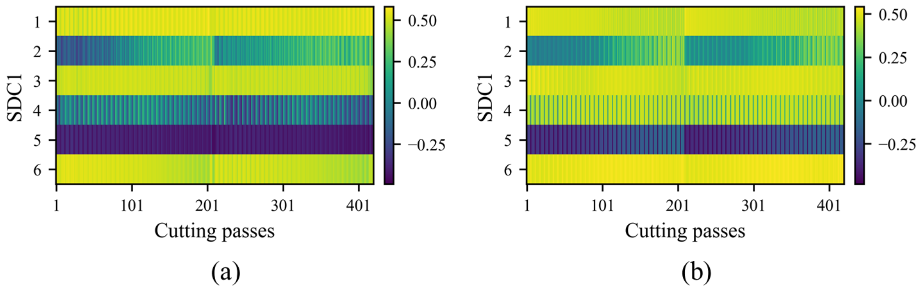


Figure 3.8: Heatmap of SDC1s vs. cutting passes: (a) under BL condition; (b) under TW condition.

Classification performance of the TDC1 and SDC1 features

In this study, the complete data set is divided into training and testing sets. Then the training set is further divided into training and validation subsets to perform K-fold cross-validation. Running the training-validation process for K rounds allows a more accurate and robust assessment of the performance of the model. As shown in Figure 3.9, the data collected from W1 and W3 is used for model training and the data from another two workpieces is used as the testing set.

This research uses a SVM classifier with the radial basis function (RBF) kernel to identify new and worn tool condition. Two parameters with the model need to be determined, which are penalty parameter and the length scale of the kernel [147]. The grid search approach with a 5-fold CV is applied to tune the parameter values. Once the two parameters have been determined, the whole training set is used to train a SVM classifier, which is then applied to the testing set.

To evaluate the classification performance of the trained model, the receiver operating characteristics (ROC) curve is plotted, which displays the true positive rate (TPR) against the false positive rate (FPR) at different classification thresholds. The closer the curve is to the top-left corner of the ROC space, the better performance the classifier achieves. The AUC is widely used to quantify the classification ability of the model. The TDC1 and SDC1 features are used separately and together as the model inputs. Figure 3.10 (a) shows the ROC curves on the testing set with TDC1s, SDC1s, and TDC1s&SDC1s, respectively. The AUCs of TDC1s and

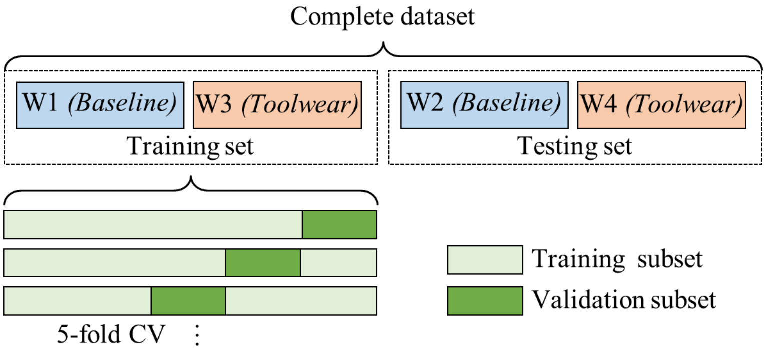


Figure 3.9: Illustration of data set splitting.

TDC1s&SDC1s are equal to 1 and that of SDC1s is close to 1, which means both positive and negative samples have been classified accurately.

The red circle on the ROC curve stands for the optimal operating point (OOP) selected by maximising Youden's J index, which is calculated as (sensitivity+specificity-1) [148]. In binary classification, the sensitivity is given by TP/(TP+FN) and specificity is TN/(TN+FP), where TN, FP, FN, TP represent true negative, false positive, false negative, true positive, respectively. The overall accuracy is (TP+TN)/(TP+FN+TN+FP). Through adjusting the classification threshold, a trade-off between sensitivity and specificity can be made. In Figure 3.10 (a), for TDC1s and TDC1s&SDC1s, the sensitivity and specificity at the OOP of ROC curve are both 100% and for SDC1s are 98.10% and 99.52%, respectively.

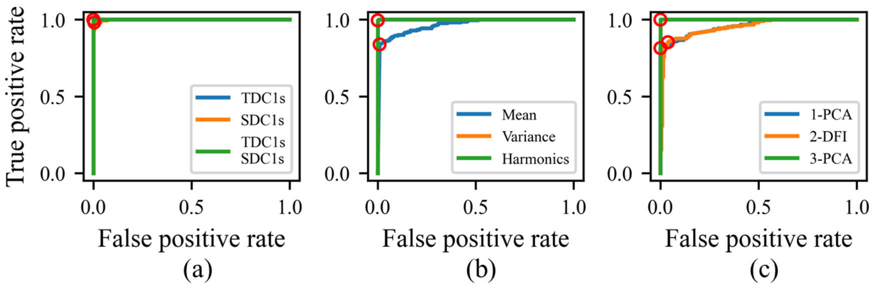


Figure 3.10: ROC curves of the trained classifiers based on different features on the testing set: (a) time and sensor domain features, (b) typical signal features, (c) features obtained by feature compression methods.

Comparison with raw signal feature-based methods

The comparative study with raw signal feature-based methods has been conducted. In this study, 13 common features in time and frequency domains are extracted from each channel of signal. The mean, variance, and spectral amplitudes at the first four harmonics of TPF are further selected as typical signal features [8]. Besides, three existing feature-layer dimension

Table 3.3: Classification results in straight-line milling experiment

Features	CV AUC	Testing AUC	Sens %	Spec %	Acc %
TDC1s	1.000	**1.000**	**100.00**	**100.00**	**100.00**
SDC1s	1.000	0.999	98.10	99.52	98.81
TDC1s SDC1s	1.000	**1.000**	**100.00**	**100.00**	**100.00**
Mean	1.000	0.969	83.81	99.05	91.43
Variance	1.000	1.000	99.52	100.00	99.76
Harmonics	1.000	1.000	99.52	100.00	99.76
1-PCA	0.980	0.959	81.43	100.00	90.72
2-DFI	0.968	0.951	85.24	96.19	90.72
3-PCA	1.000	**1.000**	**100.00**	**100.00**	**100.00**

*Sens: sensitivity, Spec: specificity, Acc: accuracy.

reduction methods are used to extract 6 dominant features from the pool of candidate features. These algorithms include principal component analysis (PCA)-based feature selection (denoted as 1-PCA) [87], dominant feature identification (DFI)-based feature selection (denoted as 2-DFI) [85], and PCA-based feature fusion (denoted as 3-PCA) [96].

Similar to Figure 3.9, features of W1 and W3 are used as training set and W2 and W4 are used for testing. Figure 3.10 (b) and (c) show the testing ROC curves based on raw signal features. In this case, using variance, harmonics or 3-PCA features can make the same perfect classification as TDC1s, while the others lead to worse classification performance.

The classification metrics, including the average AUC on the validation subsets, are summarised in Table 3.3. It is found that the TDC1/SDC1 features can generate ideal classification performance, and the same for some features like variance, harmonics, and 3-PCA. However, the testing AUC is smaller than the CV AUC with mean, 1-PCA, and 2-DFI because the distribution of these features in training and testing sets changes. In these cases, the sensitivity is between 80%~85%, which indicates around one fifth TW samples have not been correctly classified. It should be noted that 100% classification accuracy is hard to achieve in most scenarios as practical machining is quite complicated.

3.3.3 Case 2: milling process with multi-shaped features

Experimental setup

To fairly demonstrate the superiority of the proposed TDC/SDC feature-based method, a second case study was carried out on the same machine tool, but with hard-machined titanium workpiece and multi-shaped cutting features, as shown in Figure 3.11. Four 100×100×40 mm plates were machined to the designed part that included the circle-shaped, diamond-shaped and square-shaped features. A shoulder milling cutter with 7 inserts was used in the experiment. Similar to experiment 1, two workpieces were machined by brand new cutters and another two by worn cutters. The spindle speed during the machining was 300 rpm, which meant the TPF was 35 Hz. In this case, four channels of signals, which were X/Y-axial vibration on the table and X/Y-axial vibration on the spindle, were collected and analysed to extract useful features. The data set splitting and classifier training were the same as the first case study.

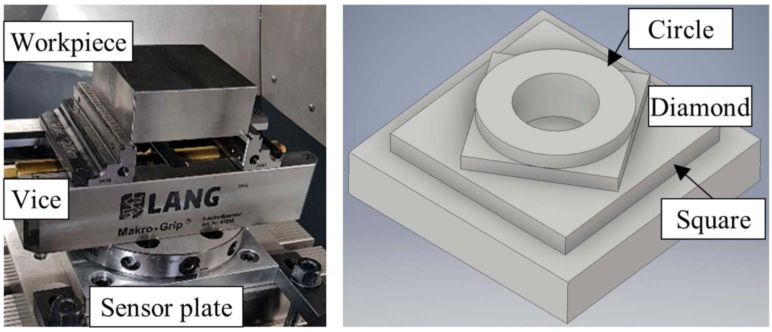


Figure 3.11: Experimental setup and the multi-shaped features.

Diagnosis results

Table 3.4 shows the classification results in all cases. The best testing AUC value is obtained with 1-PCA, SDC1s and 3-PCA for the three machining scenarios, respectively. Even though the classifiers based on the proposed time and sensor domain features do not perform the best in some cases, they still provide reasonably good AUC value and trade-off between the sensitivity and specificity. For example, in the circle milling experiment, the sensitivity and specificity of SDC1s are 61.11% and 83.33%, which are close to the best results with 1-PCA. While none of the raw signal features shows acceptable classification results in all three scenarios, for example, 1-PCA ranks first in the circle-shaped milling group while the AUC values in the other two groups are only around 0.5.

Besides, the CV AUC and testing AUC are close to each other when using TDC1/SDC1 features, which is not the situation for the raw signal feature-based methods. It indicates that the distribution of the proposed features on the training and testing sets are similar. While the raw signal features and the so-called dominant features have not shown this kind of consistency due to the influence of environmental noises. This property limits the applicability of the raw signal feature-based methods in complicated machining practice. In this sense, the proposed time and sensor domain features can be easily applied to different machining tasks with acceptable accuracy.

The performance of the proposed features is also evaluated using other classifiers. Figure 3.12 compares the classification accuracy based on TDC1/SDC1 features and raw signal features using five different classification models, which include the SVM with RBF kernel, the SVM with linera kernel, the k-nearest neighbors (KNN), the random forest (RF), and the Gaussian process classification with RBF kernel (GPC) [149, 150]. It shows that the difference in classification accuracy achieved using different features is not obvious when using different classification models. But the classification performance of the proposed time and sensor domain features is always one of the top three no matter which classifier is used. In contrast, 3-PCA based signal feature can obtain remarkable accuracy in square-shaped milling case but almost cannot distinguish different tool conditions in diamond-shaped milling.

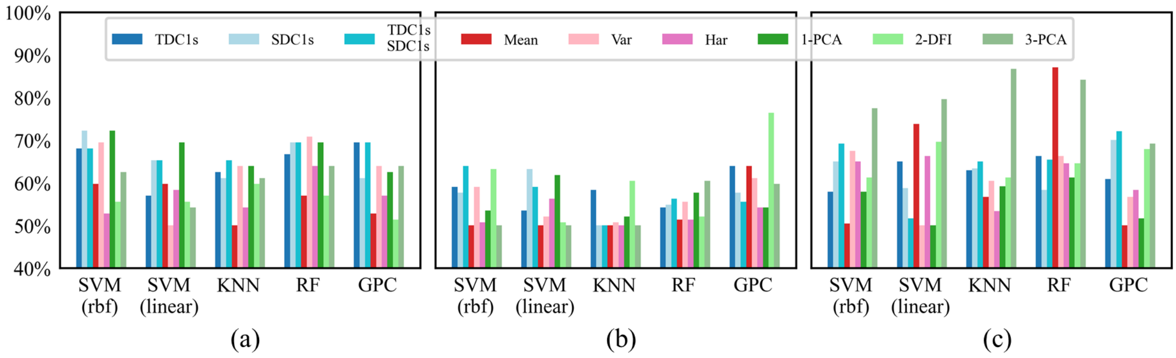


Figure 3.12: Comparison of classification accuracy using different classifiers: (a) circle-shaped feature milling, (b) diamond-shaped feature milling, (c) square-shaped feature milling.

3.3.4 Discussion

The key step in the time-sensor decomposition algorithm is estimating the covariance matrix $\mathbf{C}$. To ensure an accurate estimation, the multi-sensor data $\mathbf{X}$ should meet the following requirements.

1. Independent and identically distributed: the samples $x_1, x_2, \ldots, x_n$ are assumed to be

Table 3.4: Classification results in the multi-shaped feature milling experiment

Features	Circle				Diamond				Square			
	CV AUC	Testing AUC	Sens %	Spec %	CV AUC	Testing AUC	Sens %	Spec %	CV AUC	Testing AUC	Sens %	Spec %
TDCIs	0.687	0.666	72.22	63.89	0.704	0.511	34.72	80.56	0.844	0.651	56.67	73.33
SDCIs	0.710	0.668	**61.11**	**83.33**	0.826	**0.595**	**50.00**	**68.06**	0.894	0.520	53.33	62.50
TDCIs SDCIs	0.636	**0.696**	72.22	66.67	0.725	0.528	**80.56**	**47.22**	0.894	**0.734**	**65.00**	**73.33**
Mean	1.000	0.434	19.44	86.11	1.000	0.579	33.33	94.44	0.854	0.284	100.00	0.83
Variance	0.687	**0.703**	58.33	80.56	0.662	0.525	73.61	44.44	0.937	0.658	38.33	96.67
Harmonics	0.759	0.583	61.11	61.11	0.520	0.292	95.83	5.56	0.723	0.678	66.67	63.33
1-PCA	0.674	**0.711**	**63.89**	**80.56**	0.745	0.467	94.44	12.50	0.922	0.509	89.17	26.67
2-DFI	0.527	0.534	30.56	80.56	0.849	0.542	77.78	48.61	0.928	0.656	44.17	78.33
3-PCA	0.981	0.541	25.00	100.00	1.000	0.320	0.00	100.00	0.997	**0.808**	**67.50**	**87.50**

*Sens: sensitivity, Spec: specificity.

independent and identically distributed (i.i.d.), which means each sample is independent of any other sample and comes from the same probability distribution. Violations of the i.i.d. assumption can lead to biased estimates of $\mathbf{C}$. Unfortunately, real-world datasets often exhibit some degree of dependence or heterogeneity. Some statistical tests including Ljung-Box test and Quantile-Quantile plots, can be used to determine if the assumption is significantly violated. If this is the case, additional adjustments may be necessary to handle dependencies in the dataset, for example, using specialised covariance models or incorporating lag terms into the analysis. Additionally, domain-specific knowledge can further help to determine if the i.i.d. assumption is reasonable.

2. Homogeneity and synchronisation: the data from each sensor should have the same scale and be synchronised. The data homogeneity is ensured by standardising each variable to have a mean of zero and a standard deviation of one. The data synchronisation can be realised by leveraging both hardware and signal processing techniques. If the signals are measured by the same data acquisition system, the synchronisation is guaranteed naturally. If the signals are collected by different systems and at different sampling rates, the offset detection and re-sampling should be applied to synchronise the data. Another issue leading to data misalignment is with missing values, which should be handled consistently across all sensor by appropriate method, such as deletion, imputation, or interpolation. It is suggested to complete all these operations in the data pre-processing stage.

3. Linearity: the decomposition assumes the relationship between variables is linear. If a significant nonlinearity exists, the covariance matrix may not fully capture the underlying structure of the data. In such cases, nonlinear techniques or preprocessing methods may be needed.

4. Sample size: the number of samples should be large enough compared to the number of variables, i.e., $n \gg m$. Having more samples enables more accurate estimate of the covariance matrix. In industrial process monitoring, this can be guaranteed easily as data are collected at high sampling rate. Another issue concerning the sample size is the process dynamics. Given that the real process is always changing and results in changes in data distribution, the i.i.d. assumption may be violated if the sample size is too large. Thus, the sample size should be determined by considering the statistical requirements and process stationarity together.

Carefully pre-processing and evaluating the multi-sensor data with respect to these requirements can ensure that the data is suitable for the subsequent analysis. In addition, involving domain experts' knowledge can enhance the reliability of the decomposition results in specific scenarios.

3.4 Conclusion

In this chapter, a novel time and sensor domain data decomposition and analysis framework is proposed from the time-space decomposition method in DPSs modelling. Under this framework, the KLD algorithm is applied to decompose the multi-sensor signals into the time (TDC) and sensor-domain (SDC) components such that the original data are effectively compressed with dominant information remaining. Time series analysis including STL decomposition, AR modelling, and PSD estimation are used to extract condition-sensitive features from TDCs. Then the time and sensor domain features are used to build a fault diagnosis model, whose performance is demonstrated through milling experiments. Workpieces with different materials and shapes are machined in these experiments such that the performance of the proposed method can be evaluated comprehensively. Results from several case studies verify that the TDC and SDC features can provide at least one of the top three good classification performance in all cases. But the raw signal features that rank the top in one process may totally lose the classifying ability in another process. Meanwhile, due to the robustness to the measurements error, the proposed time and sensor decomposition framework shows a much better generalisation capability. In the following chapters, how to integrate the time and sensor domain components effectively in process monitoring will be presented.

4

Process anomaly detection based on time-sensor domain synthesis and application to tool breakage detection

4.1 Introduction

Process anomalies can happen suddenly and lead to serious accidents. An anomaly detection system can trigger alarms upon detecting faults, thereby preventing catastrophic damage to equipment. The classification-based fault diagnosis method introduced in Chapter 3 can be applied for anomaly detection by training a classifier with both normal and abnormal samples. But it is challenging to obtain a comprehensive training data set that covers various faulty conditions. Considering process faults usually cause abrupt changes in the measurements, a natural idea for anomaly detection is to characterise the measurements of healthy conditions and determine whether a new sample significantly deviates from the distribution of normal samples. Therefore, the key issue in data-driven anomaly detection becomes the description of healthy states. A practical system often relies on empirical knowledge, which is not reliable in complicated processes. This chapter focuses on TSD-based process anomaly detection, where the time and sensor domain information are utilised for feature extraction and adaptive baseline determination, respectively.

Taking tool breakage detection (TBD) as an example, timely and accurate TBD is crucial to avoid catastrophic accidents during cutting processes as mentioned in [6]. Unlike progressive wear, tool breakage often occurs suddenly without early warning because it can happen at any time of the cutting process and at any position of the tool. The purpose of TBD is to identify anomalies in real-time such that the machining can be stopped as soon as possible even before tool breakage happens. In end milling, tool breakage consists of two categories, i.e., flute breakage and shank breakage [23]. The former indicates slight chipping or tip crumble happening on cutting edges. While the latter means the tool is totally broken due to the heavy load. Considering that flute breakage is sometimes too subtle to be detected easily, this study mainly focuses on this type of anomaly.

A typical TBD system is based on a data-driven framework that aims to dig for tool condition-related information from sensor signals. The monitoring procedure usually contains three steps, namely signal collection, feature extraction, and decision making [8]. Various sensors have been used to capture raw data in real-time, including the spindle current [151–153], cutting force [154, 155], vibration [48] and acoustic emission (AE) [23, 156]. Although most existing TBD systems are single-signal dominated, the multi-sensor fusion strategy is gradually applied to this task as the tool status can be reflected more comprehensively [6, 25, 48]. From raw signals, multiple features are extracted in the time domain, frequency domain, and time-frequency domain [56, 157]. These features can be used to quantify the change in signals caused by tool breakage. To obtain the most representative features, feature selection or fusion is performed manually or via some dimension reduction algorithms [86, 87, 98]. Deep learning models can directly mine tool state information from raw data, thus the application of deep learning has emerged in recent years [158]. For TBD, constructing a sensitive feature from a single sensor data is still the mainstream approach, which is not robust enough to condition changes and environmental noises. Hence, the integration of multiple sensors and multiple features deserves more investigation.

Earlier TBD systems make decisions based on a threshold. For example, in [156], the Mahalanobis distance from the feature vector to the reference feature set collected under normal conditions was used to indicate the tool breakage state. If the current distance value exceeded a threshold value, the tool was considered running in abnormal conditions. To improve the robustness of the TBD, some studies try to adopt the floating threshold strategy, which automatically adapts to changes in cutting environments. In [151], the concurrent learning approach was used to adjust the tolerance band in real-time. Thus this method could be adjusted to changing conditions and avoid false alarms. The threshold-based method is straightforward and widely used in commercial TBD systems [155]. Recently, the TBD is considered a binary classification problem and the machine learning model provides another solution to TBD. Researchers have applied convolutional neural networks to perform anomaly detection of cutting tools [153]. Only normal samples were used for training, and the loss values with testing samples as inputs were selected as tool health indices. Additionally, some studies have attempted to develop supervised classification models which use generative adversarial networks-based data augmentation [23] or classifier improvement [152] to address the tool breakage data imbalance problem. However, the successful implementation of these methods in different cutting processes also relies on expert knowledge, which is not always available and reliable. In summary, the current TBD technique still faces some challenges as follows.

Firstly, the use of multi-sensors and multi-domain features enlarges the number of monitoring variables. Although the feature-layer dimension reduction technique can generate representative features, this method cannot be implemented directly when machining processes or cutting conditions change. For instance, the loading matrix used in principal component analysis should be recalculated in different processes. Besides, the feature-layer fusion inevitably causes a loss of sensor-domain information. Thus, a straightforward but effective solution is needed to address challenges associated with processing high-dimensional multi-sensor data for TBD.

Secondly, the limited accessibility of training data hinders the effort to improve the generalization capability of decision-making models. A large amount of broken samples are necessary to set an appropriate threshold or train a robust classifier. However, considering the costs and complexities of machining, it is unrealistic to perform extensive destructive experiments to obtain tool breakage data. On the other side, unsupervised methods avoid model training and make decisions according to the data distribution directly. Thus they can be easily adopted under changing conditions, which is significant to industrial applications. But a key issue with unsupervised models is how to determine a reasonable baseline state. Nowadays, most systems still depend upon the operator's empirical knowledge. Consequently, there is still a considerable gap between unsupervised decision-making models and practical applications.

To resolve these challenges, the present study proposes a new unsupervised approach for tool breakage detection based on the time-sensor domain data decomposition and analysis framework. The basic idea is to decompose multi-sensor data into separate time and sensor domains to compress the raw information into a significantly smaller set of time and sensor domain data. Then, an unsupervised learning technique is introduced to detect tool breakage in real-time. The main contributions are,

1. The issue of high-dimensional features is addressed by the proposed TSD strategy, realising the dimension reduction in the data layer as mentioned in Chapter 3. Both time and sensor domain information are helpful to detect tool breakage accurately. This study is the first attempt to apply the concept of TSD to multi-sensor data analysis for TBD.

2. A novel time-sensor domain synthesis framework is designed to provide a solution to unsupervised detection of cutting tool breakage without the involvement of a training dataset. The sensor domain information is exploited to determine the baseline passes and adaptively update the TBD threshold while the time domain information is used to generate features for TBD. The approach can resolve difficulties with lacking sufficient data to train a robust supervised classification model for TBD, and circumvent problems with existing unsupervised TBD techniques in threshold determination.

3. The proposed approach has been applied on multi-sensor data from two milling experiments to investigate its effectiveness and applicability in different machining operations. The results show that multi-sensor signals can be compressed and then used to perform an unsupervised detection of breakage with cutting tools.

The proposed time-sensor domain synthesis framework is applicable to anomaly detection of many industrial processes. The property of time and sensor domain components is firstly revealed and applied in this study. In the following sections, the proposed algorithm and experimental results will be introduced. Afterwards, some practical issues of this method will be discussed to enhance its potential applicability in other fields.

4.2 Methodology

Figure 4.1 shows the flowchart of the proposed unsupervised tool breakage detection approach, consisting of four steps, which are: time-sensor domain data decomposition, TDCs analysis, SDCs clustering, and anomaly detection. The whole cutting process of a tool consists of several cutting passes. One pass refers to one continuous machining process, for example, the tool moving along the workpiece edge once or a circular path once can be called a cutting pass. Once the multi-sensor signals of one cutting pass are available, the first step is to pre-process the raw data and decompose them into TDCs and SDCs via the KLD algorithm. Secondly, the time domain features are extracted from TDCs through STL-based time series decomposition, AR modelling, and PSD estimation. Note that the current steps are the same as introduced in Chapter 3. Next, the SDC vectors are clustered using a sequential clustering framework. Based on the clustering results of all available SDCs from cut 1 to the current pass, the baseline passes, the baseline space, and outlier threshold are determined. Afterwards, the Mahalanobis distance (M-Dist) of the current TDC feature vector to the baseline space is calculated and compared against the threshold. If the M-Dist value exceeds the threshold, a tool breakage event could happen during this cutting pass. Details of this approach are described as follows.

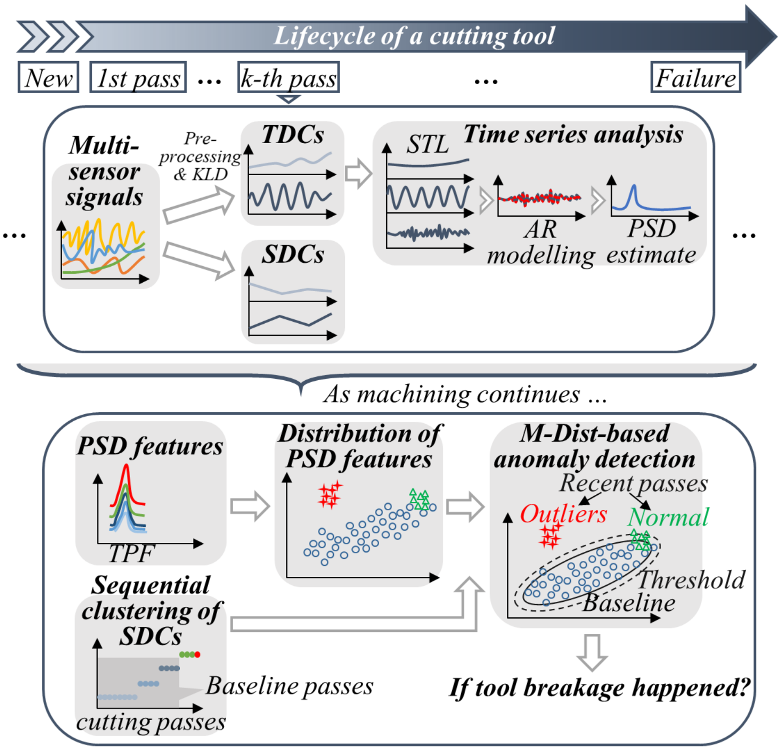


Figure 4.1: Flowchart of the proposed unsupervised tool breakage detection system.

4.2.1 Key idea

Figure 4.2 illustrates the key idea of anomaly detection algorithm, that is, to define a baseline space and determine whether a new sample exceeds its boundary. The baseline space represents the distribution of normal samples, and new samples located outside the baseline space indicate faulty conditions. As shown in the figure, selecting an appropriate baseline space is crucial; a conservative selection may cause a normal sample to be marked as an outlier, while a radical selection could lead to the miss of a true abnormal sample.

Previous methods estimate the baseline conditions based on empirical experiences, which

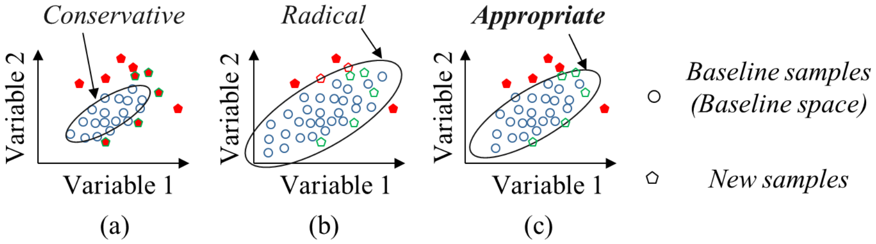


Figure 4.2: Anomaly detection results using different kinds of baseline definition: (a) conservative; (b) radical; (c) appropriate, assuming one sample can be represented by two feature variables. The indication of the colours: green outline: normal; red outline: abnormal; red filled: detected anomaly.

is often not reliable enough [25]. Because equipment often experiences performance degradation and faults can occur randomly at any stage. Incorrectly defining healthy states can lead to false alarms or missed detections. Under the proposed TSD framework, SDCs can aid in baseline selection. Given that SDCs capture time independent but sensor nature dependent information of the machining process, notable changes in SDC vectors can result from change in cutting process, including tool condition as shown in Chapter 3. Therefore, a novel time and sensor domain synthesis approach is proposed to estimate baseline samples by employing clustering analysis for SDCs and subsequently detect anomalies using the time domain features extracted from TDCs. As this baseline selection is made and updated adaptively, the tool breakage detection can be performed in real-time and adapt to normal degradation processes, without involving labelled samples as in most existing TBD techniques.

4.2.2 Sensor domain data clustering

The TSD algorithm and TDC feature extraction have been presented in Chapter 3 and will not be repeated here. For the sensor domain data analysis, a novel sequential clustering algorithm is developed according to [159]. Let the i-th SDC of the l-th cutting pass be denoted as a vector $\boldsymbol{\varphi}_i^l \in \mathbb{R}^n$. Concatenating the n' dominant SDCs of the l-th cutting pass produces an nn'-dimensional vector $\boldsymbol{\varphi}^{l^T} = \left[\boldsymbol{\varphi}_1^{l^T} \dots \boldsymbol{\varphi}_{n'}^{l^T}\right]$. The distance between $\boldsymbol{\varphi}^{l_1}$ and $\boldsymbol{\varphi}^{l_2}$ can be used as an indicator of the similarity between the condition of cutting passes l_1 and l_2. To implement the analysis in real-time, the sequential K-means clustering method, where the Euclidean distance is considered, is proposed to analyse SDC vectors sequentially as shown in **Procedure 1**.

In this procedure, it is assumed that during the first l_{init} passes, the tool is in good condition. The parameter R is initialised according to the distribution of the first l_{init} SDC vectors. This parameter determines whether a new SDC vector should be assigned to an existing cluster or

Procedure 1 Sequential clustering of SDC vectors

1. **Initialise the first cluster:** Take the components in cluster 1 as the concatenated SDCs of the first l_{init} cutting passes and calculate the centroid of cluster 1 as $\boldsymbol{c}_1 = \frac{1}{l_{init}} \sum_{i=1}^{l_{init}} \boldsymbol{\varphi}^i$.
2. **Determine cluster bound *R*:** Calculate the Euclidean distance between any two components in cluster 1: $\mathrm{d}_{ij} = \mathrm{d}\left(\boldsymbol{\varphi}^i, \boldsymbol{\varphi}^j\right), \quad \forall i, j \in \{1, 2, \ldots, l_{init}\}$, obtain $R = \mu_\mathrm{d} + 3\sigma_\mathrm{d}$, where μ_d and σ_d denote the mean and standard deviation of these distances, and let $l = l_{init},\ K = 1,\ N_K = l_{init}$.
3. **Allocate $\boldsymbol{\varphi}^{l+1}$, the SDC of the l+1-th cutting pass, to an existing or new cluster:**
   - 3.1 Calculate $\mathrm{d}\left(\boldsymbol{\varphi}^{l+1},\ \boldsymbol{c}_k\right)$, for $k = 1, \ldots, K$ and find $k = k^*$ such that $\mathrm{d}\left(\boldsymbol{\varphi}^{l+1},\ \boldsymbol{c}_{k^*}\right)$ is the smallest.
   - 3.2 If $\mathrm{d}\left(\boldsymbol{\varphi}^{l+1},\ \boldsymbol{c}_{k^*}\right) \leq R$: assign $\boldsymbol{\varphi}^{l+1}$ to cluster k^*, update N_{k^*} as $N_{k^*} + 1$, and $\boldsymbol{c}_{k^*}$ as $\boldsymbol{c}_{k^*} + \frac{1}{N_{k^*}}(\boldsymbol{\varphi}^{l+1} - \boldsymbol{c}_{k^*})$.
   - 3.3 If $\mathrm{d}\left(\boldsymbol{\varphi}^{l+1},\ \boldsymbol{c}_{k^*}\right) > R$: create cluster *K*+1 with $N_{K+1} = 1,\quad \boldsymbol{c}_{K+1} = \boldsymbol{\varphi}^{l+1}$, and update the parameter *R* as $\max\left(R, \max\left(\mathrm{d}\left(\boldsymbol{c}_i, \boldsymbol{c}_j\right),\ \forall i, j \in \{1, 2, \ldots, K+1\}\right)\right)$.
4. **Update l as l+1 and return to 3 until all cutting passes have been completed.**

a new cluster. When a new cluster is created, *R* is updated to accommodate the distribution of the clusters of available SDC vectors.

After performing the sequential clustering analysis for the SDC vectors using **Procedure 1**, the baseline cutting passes are determined and updated whenever a different cluster appears. The details are shown in **Procedure 2**.

Here the change in clustering results indicates the possible change in cutting tool conditions. Based on the baseline cutting passes determined using **Procedure 2**, the TDC features will be used for the detection of cutting tool breakage.

4.2.3 Anomaly detection

The anomaly detection is performed by comparing the Mahalanobis distance of the TDC feature of a new cutting pass in the Mahalanobis space with an outlier threshold *T*. If the M-Dist value exceeds the threshold, an abnormal event like tool breakage could very likely happen during this cutting pass and the machining should be stopped immediately.

Procedure 2 **Baseline cutting pass determination and updating**

1. **Initialise the baseline passes:** Assume the SDC of the l_0-th cutting pass, φ^{l_0}, is the first component allocated to cluster 2, determine the initial baseline passes as cutting passes 1~$l_0 - 1$, let $l = l_0$ and assign the cluster number of the l-th cutting pass as $k_l = 2$.

2. **Update the baseline passes according to cluster k_{l+1} which is the cluster that φ^{l+1} belongs to:**

   2.1 If $k_{l+1} = k_l$, i.e., φ^{l+1} is in the same cluster as the cluster of φ^l, then the baseline passes remain unchanged.

   2.2 If $k_{l+1} \neq k_l$, i.e., φ^{l+1} is in a cluster different from the cluster of φ^l, then update the baseline passes as cutting passes 1~l

3. **Update l as $l + 1$, and return to 2 until all cutting passes have been completed.**

The Mahalanobis space consists of the TDC feature set of the baseline passes determined by **Procedure 2**. To improve the robustness of the proposed approach and determine an appropriate threshold T, a shrinkage technique is applied and an improved Monto Carlo algorithm is developed. The details can be summarised in the following steps:

Step 1 (Mahalanobis space construction): Assume TDC1 and SDC1 are dominant pairs, the TDC1 feature vector can be formed as $\boldsymbol{f} = [S_1(f_1) \ldots S_1(f_t)]^T$, where $f_1, \ldots, f_t$ are frequencies of concern, with t denoting the number of frequency points. Based on the clustering results of SDC1s, the baseline passes can be determined as, for example, cuts 1~l^*. The Mahalanobis space is then constituted as a matrix $\mathbf{F}_{BL} = [\boldsymbol{f}_1, \ldots, \boldsymbol{f}_{l^*}]$ with $\boldsymbol{f}_i$ representing the TDC1 features of cutting pass i, $i = 1, \ldots, l^*$. The M-Dist of a new feature vector $\boldsymbol{f}_{l'}$ is given by

$$\mathrm{D}(\boldsymbol{f}_{l'}) = \sqrt{\left(\boldsymbol{f}_{l'} - \boldsymbol{\mu}_{BL}\right)^T \mathbf{S}_{BL}^{-1} \left(\boldsymbol{f}_{l'} - \boldsymbol{\mu}_{BL}\right)} \tag{4.1}$$

where $\boldsymbol{\mu}_{BL}$ and $\mathbf{S}_{BL}$ denote the mean and covariance matrix of $\mathbf{F}_{BL}$, respectively. However, $\mathbf{S}_{BL}$ can be ill-conditioned when l^* is relatively smaller than the dimension t or any two columns in $\mathbf{F}_{BL}$ are highly correlated [160]. An inversion of an ill-conditioned covariance matrix causes numerical instability, which means a small change in the feature vector can lead to large change in M-Dist, making the obtained M-Dist unreliable. To resolve this problem, the shrinkage-based technique is utilised to shrink $\mathbf{S}_{BL}$ towards the identity matrix by

$$\mathbf{S}_{BL}^* = (1 - \alpha)\,\mathbf{S}_{BL} + \alpha \mathbf{I}_t \tag{4.2}$$

where $\alpha \in [0, 1]$ is the shrinkage intensity and $\mathbf{I}_t$ is identity matrix. Compared with $\mathbf{S}_{BL}$, the eigenvectors of $\mathbf{S}_{BL}^*$ are unchanged but the eigenvalues are better conditioned. The shrinkage intensity α is obtained according to the desired condition number κ by

$$\alpha = \frac{\lambda_1 (\mathbf{S}_{BL}) - \lambda_t (\mathbf{S}_{BL}) \kappa}{\kappa - 1 + \lambda_1 (\mathbf{S}_{BL}) - \lambda_t (\mathbf{S}_{BL}) \kappa} \tag{4.3}$$

where $\lambda_1 (\mathbf{S}_{BL})$ and $\lambda_t (\mathbf{S}_{BL})$ are the maximal and minimal eigenvalues of $\mathbf{S}_{BL}$.

Step 2 (Threshold determination): In general, the outlier threshold is determined by calculating the quantiles of Chi-squared distribution or using the Monte Carlo algorithm [161, 162]. However, to ensure the threshold is robust enough to the measurement errors or noises, the Monte Carlo algorithm presented in [162] is improved to determine the outlier threshold in extreme situations as follows,

1. Assume the baseline passes contain cuts 1~l, generate l samples randomly from a multivariate normal distribution specified by the $\boldsymbol{\mu}_{BL}$ and $\mathbf{S}^*_{BL}$;

2. Calculate M-Dist values of all generated samples and record the largest M-Dist;

3. Repeat steps 1) and 2) for 1000 trials, sort all maximum M-Dist values, and determine the threshold T above which 1% of trails occurs.

Step 3 (Anomaly detection): The M-Dist is calculated using the shrunken covariance matrix $\mathbf{S}^*_{BL}$ by

$$\mathrm{D}^*(\boldsymbol{f}_{l'}) = \sqrt{(\boldsymbol{f}_{l'} - \boldsymbol{\mu}_{BL})^T \mathbf{S}^{*\,-1}_{BL} (\boldsymbol{f}_{l'} - \boldsymbol{\mu}_{BL})} \tag{4.4}$$

and is compared with the threshold T. If M-Dist is larger than T, then an alarm of the potential breakage risk is triggered and a careful check is needed. A flowchart for the 3-step procedure is shown in Figure 4.3. When a new cutting pass is completed, multi-sensor data are collected and the corresponding TDC feature becomes available. At the same time, the baseline passes have already been determined according to the clustering analysis of SDCs from all previous cutting passes, which means $\boldsymbol{\mu}_{BL}$, $\mathbf{S}^{*\,-1}_{BL}$, and T have already been known. The M-Dist of the new sample can be calculated and compared with T to detect the corresponding tool condition. The update of baseline passes follows the rule presented in **Procedure 2**. Whenever the baseline is updated, the Mahalanobis space and the outlier threshold need to be recalculated as shown in this section. This algorithm is carried out until tool breakage is detected or machining is completed.

It is worth noting that the shrinkage of the covariance matrix $\mathbf{S}_{BL}$ is necessary. Otherwise, the primal Mahalanobis space is oversensitive to noises such that a small perturbation in the feature vector $\boldsymbol{f}_{l'}$ could be amplified significantly, which generates a large M-Dist value. In this sense, any new feature vector might be marked as an outlier condition. The shrinkage

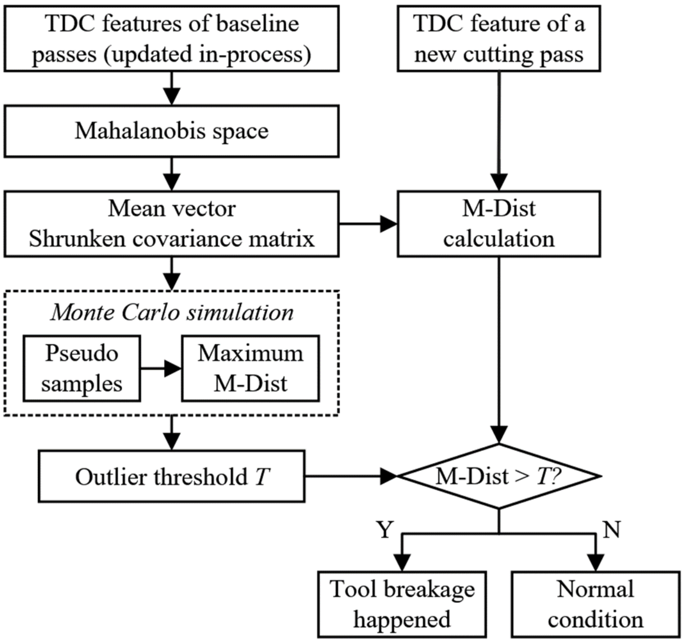


Figure 4.3: Flowchart of the Mahalanobis distance-based anomaly detection algorithm.

technique, intuitively, adjusts the baseline space by compressing the directions of large eigenvectors and stretching the directions of small eigenvectors [163]. This helps to reduce the estimation error and improve the accuracy and stability of M-Dist calculation. Hence, the anomaly detection performance can be improved.

Regarding the threshold determination, the authors in [162] generate random samples from a zero mean and unit standard deviation normal distribution. This method ignores the correlation structure of variables and could yield an inapplicable threshold and unreliable detection results. The proposed approach, namely an improved Monte Carlo, aims to simulate the possible distribution of samples in the baseline space. The distance of samples near the boundary relative to the centre is defined as the outlier threshold T. This strategy can provide a more reliable result no matter where the space centre is. Additionally, the number of trials needed for a Monte Carlo simulation depends on the specific requirements of the problem to be solved. Here 1000 trials are used considering the trade-off between the estimation stability and computational efficiency.

4.2.4 Implementation algorithm

The first step to implement the unsupervised tool breakage detection algorithm is to decompose multi-sensor data of each cutting pass into dominant TDCs and SDCs by Eqs. (3.2)-(3.6). In the time domain, the PSDs of TDCs are estimated by Eqs. (3.8)-(3.10), from which the TDC features are extracted. In the sensor domain, sequential clustering is performed on SDC vectors by **Procedure 1**. According to the clustering result, the baseline cutting passes are determined and updated through **Procedure 2**. Then the baseline Mahalanobis space defined by the shrunken covariance matrix is obtained by (4.2) and (4.3). The outlier threshold T is computed using the improved Monte Carlo algorithm. After the baseline space has been determined, the M-Dist value of the new feature vector is calculated by Eq. (4.4) and compared with T for anomaly detection.

An alarm is triggered when the M-Dist value exceeds the threshold T. In this algorithm, the tool condition is represented by TDC features which show process dynamics. The detection is always conducted with the latest baseline space that is determined according to the clustering result of SDCs. In this case, the minimum lag of the proposed TBD approach is equal to the time of one cutting pass. This is acceptable in practice to prevent furher damage to the workpiece. Assume TDC1 and SDC1 are selected as dominant components, the whole procedure of TBD is conducted as described in **Procedure 3**.

4.3 Case 1: milling process with a straight-line tool path

To validate the effectiveness of the TSD-based tool breakage detection approach, two machining experiments were conducted. The processes were face milling with straight-line and circular-line tool paths, respectively. Tool breakage happened in both experiments. Through analysing the multi-sensor data based on the proposed approach, the tool breakage events could be accurately detected. In this section, the first experiment study will be introduced.

4.3.1 Experimental setup

The experiment was performed on a 5-axis milling machine DMG MORI's NMV8000. A Sandvik CoroMill Plura 1620 solid carbide square shoulder 16 mm-diameter end mill (1P330-1600-XA 1620) with 3 flutes was used to machine a rectangular block TC4 workpiece as shown in Figure 4.4 (a). The straight-line cutting path was repeated by stepping along the direction perpendicular to the feed motion. The cutting pass was repeated 350 times during the experiment. The rotational speed and feed rate were set as 1592 rpm and 859 mm/min, respectively. The cutting width and depth were 0.5 mm and 8 mm. The data collection system is shown in Figure 4.4 (a). The accelerometer was embedded into a sensor plate and mounted beneath the workpiece. The microphone was placed close to the cutting area. The

Procedure 3 Unsupervised detection of cutting tool breakage

```
1.  Input: multi-sensor data $\mathbf{X}^l$, $l = 1, 2, \ldots L$, with initial cutting passes: cut 1~$l_{init}$
2.  Output: whether a tool breakage occurs or not
3.  for $l$ = 1 to $l_{init}$
4.     Process and standardize $\mathbf{X}^l$ to obtain $\mathbf{X}'^l$ by Eq. (3.2)
5.     Decompose $\mathbf{X}'^l$ into TDC1 $\boldsymbol{y}^l$ and SDC1 $\boldsymbol{\varphi}^l$ by Eqs. (3.3)-(3.6)
6.     Estimate $S^l(f)$ from $\boldsymbol{y}^l$ by Eqs. (3.8)-(3.10) to get a feature vector $\boldsymbol{f}^l = \left[S^l(f_1) \ldots\ S^l(f_t)\right]^T$
7.  end for
8.  for $l$ = $l_{init}$+1 to $L$
9.     Run steps 4~6 (data decomposition and TDC feature extraction) to get $\boldsymbol{\varphi}^l$ and $\boldsymbol{f}^l$
10.    if the baseline space has not been determined
11.      Sequential clustering analysis of $\boldsymbol{\varphi}^l$ by Procedure 1
12.      Run Procedure 2 as follows
13.      if $\boldsymbol{\varphi}^l$ belongs to the same cluster as the last one
14.         Continue
15.      else
16.         Update the baseline space and $T$
17.    else
18.      Run steps 11~16 (sequential clustering and baseline updating)
19.      Calculate $\mathrm{D}^*(\boldsymbol{f}^l)$ by Eq. (4.4)
20.      if $\mathrm{D}^*(\boldsymbol{f}^l) < T$
21.         Machining continues
22.      else
23.         Stop and replace the broken tool
24. end for
```

specifications of these sensors have been given in Table 3.1. In this study, the sound and tri-axial vibration signals were sampled at 10240 Hz and the number of variables was 4 in total.

During the cutting trial, a microscope was used to take images of cutting edges after each 20-30 cutting passes. Figure 4.4 (b) shows the images of flutes from the new state to broken. It can be found that the build-up edge is formed as machining goes on. Then, in the 180th cutting pass, chipping, and even tip fracture occur on flutes 2 and 3. To collect more data under faulty conditions, this tool is still used for machining until cut #350.

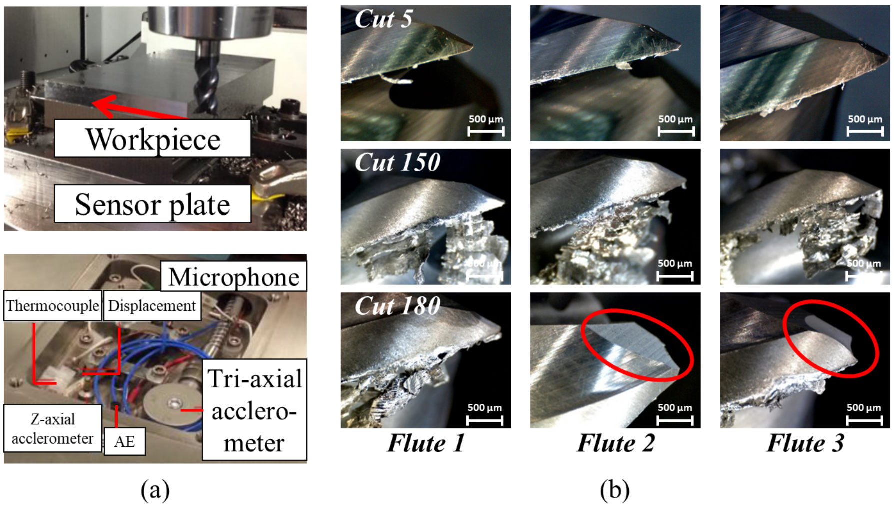


Figure 4.4: (a) Experimental setup and the sensor plate; (b) images of three flutes after cuts 5, 150 and 180.

4.3.2 Experimental results

Step 1: TSD of multi-sensor data by Eqs. (3.2)-(3.6)

In this case, the TPF is 79.6 Hz and the tool frequency is 26.53 Hz. The tool condition-related information should be around this frequency range. Thus, a band-pass filter was applied to each sensor signal over the frequency range from 20 to 90 Hz then the filtered data were down-sampled at 954 Hz, which was about 12 times the TPF. The selection of down-sampling frequency f_{ds} follows the three principles as follows,

1. f_{ds} should exceed twice the pass frequency such that the Nyquist sampling theorem is followed.

2. f_{ds} is an integer multiple of the TPF such that the period parameter in STL algorithm can be readily determined.

Table 4.1: Eigenvalues of the signals' covariance matrix, cut #1

i	1	2	3	4
Eigenvalues	2.58	1.16	0.24	0.02
CPV (%)	64.5	93.5	99.5	100.0

3. f_{ds} should not be too large, otherwise bringing about unnecessary computational costs.

The signals used for the analysis include sound and tri-axial vibration. The time-sensor decomposition was performed on these data. Table 4.1 shows the eigenvalues of the covariance matrix of the signals collected from cut #1. The first component accounts for almost 65% of the total variance of original measurements, thus TDC1/SDC1 is selected as the dominant component. The amount of data to be analysed is compressed from 4 sets of sensor signals to one time series and one 4-dimensional vector.

Step 2: Time domain features extraction by Eqs. (3.8)-(3.10)

The TDC1 is firstly decomposed into the trend, seasonal and residual components. Figure 4.5 shows the TDC1 of cut 1 and its sub-components in the time and frequency domain. The frequency spectrum of TDC1 shows three sharp peaks at 26.6 Hz, 53.1 Hz and 79.7 Hz. Furthermore, it can be found that the information of focus is at TPF of 79.6 Hz and mainly in the residual component from Figure 4.5 (b). An AR model is estimated for the residual time series. Considering only two peaks appear in the spectrum, an AR(4) model is sufficient to capture the process dynamics, which is verified by a good agreement of multi-step-ahead predicted outputs with the actual values (RMSE=0.156) as shown in Figure 4.6 (a).

Figure 4.6 (b) shows the single-sided PSDs of TDC1 residual components of different cutting passes. All spectra have a sharp peak at TPF. The patterns of PSD curves of the passes after cut #180 deviate from the others, which is an indication of the change in the machining dynamics. The spectral amplitudes at 10 frequencies from 70 to 90 Hz constitute the TDC1 feature vector.

Step 3: Sequential clustering of SDCs by **Procedure 1**

Figure 4.7 (a) shows the heatmap of SDC1s vs. cutting passes. Two distinct patterns can be found. The values of SDC1s fluctuate drastically after around cut 180. This change coincides with the experimental observation of tool breakge after cut 180. However, the SDC1s of cutting passes close to cut 150 show a similar pattern as that around cut 180, which means the tool

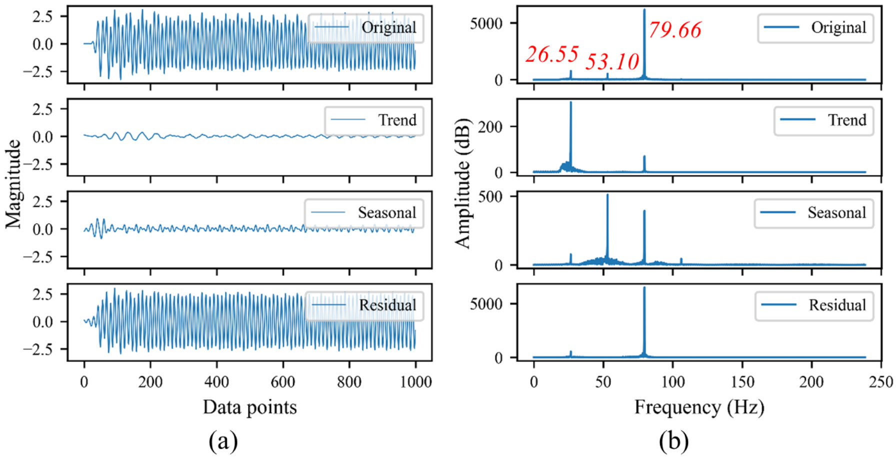


Figure 4.5: (a) TDC1 and sub-components in cut #1; (b) DFT analysis results of TDC1 and sub-components in cut #1.

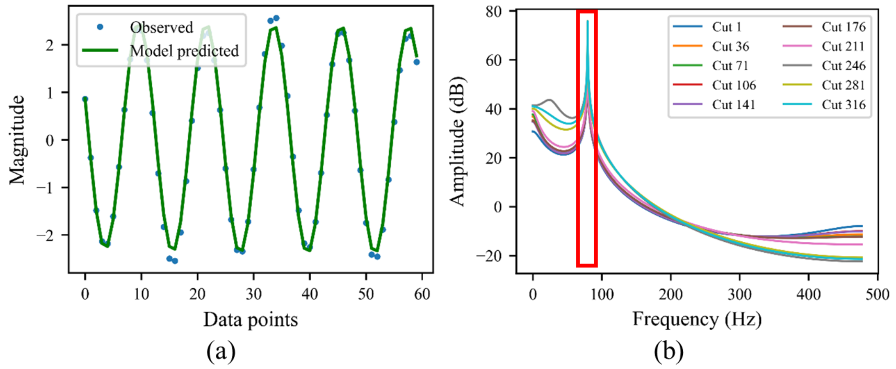


Figure 4.6: (a) AR(4) model predicted outputs vs. true values in cut #1 with RMSE 0.156; (b) the PSDs of TDC1's residual components.

breakage events cannot be identified exactly using SDC1s only. Even though, it is still possible to extract process-related information from the sensor domain.

The sequential clustering results of SDC1s are shown in Figure 4.7 (b). The first 10 passes were defined as initial passes. The second cluster does not appear till cut 142. Then the third and fourth cluster appear at cut 145 and 146, respectively. The rapid change in clustering results happens again from cut 172 to 181. The result shows that the sequential clustering analysis can quantify the divergence of SDC patterns. According to the change in clustering results, the baseline cutting passes are defined and updated following **Procedure 2**.

Step 4: Anomaly detection results using **Procedure 3**

As shown in Figure 4.7 (b), at first the baseline passes are defined as cuts 1-141. The corresponding TDC1 feature vectors constitute a Mahalanobis space with a desired condition number κ of 400 [164]. Then the 1% threshold is determined by the improved Monte Carlo

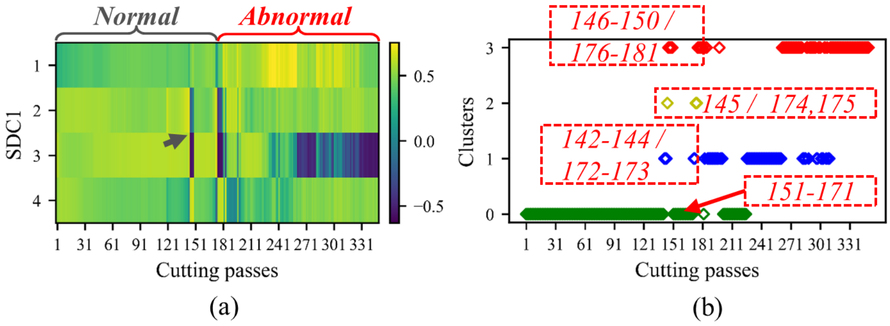


Figure 4.7: (a) Heatmap of SDC1s vs. cutting passes; (b) sequential clustering of SDC1s.

algorithm. The M-Dist values of cuts 142-144 are calculated and none of them exceeds the threshold as can be seen in Figure 4.8 (a), which means no anomaly event happens during this period. The clustering result changes again at cut 145, thus the baseline passes are updated to contain cuts 1-144. With the updated covariance matrix and threshold, anomaly detection is performed on the TDC1 feature of cut 145. In this way, the whole procedure is carried out along with the machining process. Figure 4.8 (a) shows the changes in M-Dist and 1% threshold with the adaptive adjustment of baseline passes. It can be found that the outlier condition appears at cut 179, which agrees with the experimental observation. On the other hand, without sensor domain information, the baseline passes can only be estimated based on empirical knowledge, for instance, cuts 1-50 or cuts 1-100. In this situation, wrong detection results of cut 57 or 121 are generated as shown in Figure 4.8 (b) and (c), respectively. The results indicate that in the absence of training data, the empirical estimation of baseline passes cannot detect tool breakage reliably.

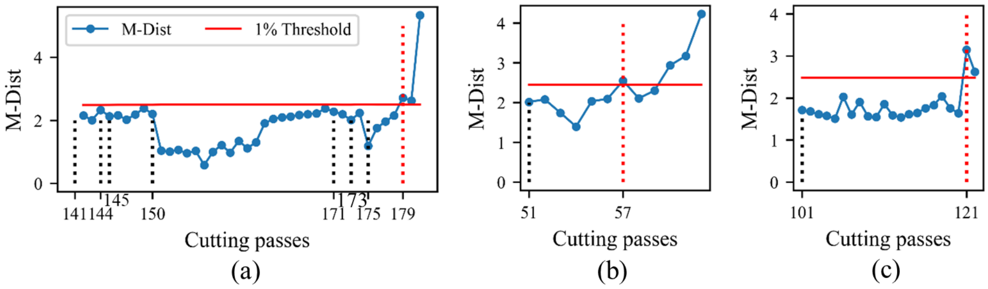


Figure 4.8: Tool breakage detection results: (a) adaptive adjustment of baseline cutting passes; manual definition of baseline passes: (b) cuts #1-50; (c) cuts #1-100.

Table 4.2 shows the value of the threshold and the M-Dist range of the passes to be detected with the updating of the baseline passes. Compared with Figure 4.8 (a), the detailed results at every stage can be clearly seen. There are no alarms reported until cut 179.

Table 4.2: Tool breakage detection results; using the proposed approach in the straight-line milling experiment

BL: #1-	T	# to be detected	M-Dist range	Detect fault	Real fault
141	2.488	142-144	2.007-2.328	×	×
144	2.490	145	2.131	×	×
145	2.493	146-150	2.025-2.385	×	×
150	2.498	151-171	0.581-2.374	×	×
171	2.495	172-173	2.011-2.195	×	×
173	2.502	174-175	1.858-2.234	×	×
175	2.498	176-181	1.758-5.333	✓#179	✓#180

4.3.3 Discussion on the algorithm design

The effectiveness of the shrinkage of covariance matrix and the improved Monte Carlo algorithm has been investigated. Figure 4.9 show the TBD results using $\mathbf{S}_{BL}$ and using the thresholds determined by Monte Carlo algorithm [162], respectively. As mentioned above, a small perturbation in the feature vector will induce a significant change in the value of the M-Dist. As a result, almost every new cutting pass is identified as an anomaly as shown in Figure 4.9 (a). In this case, the detection result makes no sense. On the other side, if random samples in Monte Carlo simulation are generated from a zero mean and unit standard deviation normal distribution, most of them spread around a position far away from the baseline samples. The determined threshold is not sensitive to the outlier relative to the baseline space. Consequently, no tool breakage event has been identified as shown in Figure 4.9 (b). Therefore, the proposed approach using the shrinkage technique and improved Monte Carlo algorithm can solve the problem with existing techniques and fulfil the task of TBD well.

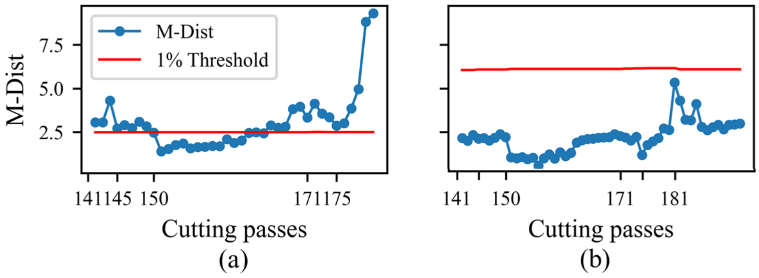


Figure 4.9: Tool breakage detection results: (a) where no the shrinkage of the covariance matrix; (b) where using Monte Carlo algorithm [162] to determine the threshold.

Regarding the efficiency of the proposed algorithm, the time-consuming of each step is recorded and analysed. It is implemented on a laptop with Intel(R) Core(TM) i5-10310U CPU and 16 GB RAM. The software is developed in Python 3.8.5 on Windows 10. The running time for time-sensor decomposition of the signal is 0.019 s; for TDC1 analysis is 0.179 s; for SDCs clustering is 0.01 s; for threshold calculation is 1.031 s; for anomaly detection is 0.002 s. Therefore, if the threshold update is not needed, it takes around 0.3 s to obtain a detection result; if the baseline is updated, it takes about one extra second to update the threshold. Given that the threshold does not need to be recalculated after every cutting pass, the efficiency is satisfactory for real-time TBD.

4.3.4 Comparative study

The performance of the proposed method is compared with four existing unsupervised approaches, which are based on fixed threshold [156], floating threshold [151], one-class support vector machine (OCSVM) [165], and ECOD [166], respectively. In the fixed threshold method, the wavelet transform (WT) is applied to construct feature vectors from raw signals. Three sets of baseline passes are used for reference, within which the 3-sigma limit of M-Dist values is defined as the threshold. For the floating threshold method, the baseline does not need to be determined in advance. Instead, the WT-based features of four signals are used. The OCSVM and ECOD are unsupervised anomaly detection algorithms. Similar to [153], four time-domain features are extracted from raw data and used for anomaly detection.

The TBD result is evaluated in terms of timeliness and overall accuracy. The former is defined by the difference between the detected breakage point (DBP) and real breakage point (RBP), i.e., $\Delta = |DBP - RBP|$. The accuracy is quantified by the fault detection rate (FDR) and false alarm rate (FAR). The total accuracy (Acc) is also given. FDR measures the proportion of anomalies correctly identified by the system among all actual anomalies. FAR is the proportion of normal samples that are incorrectly identified as anomalies out of all normal samples. Both metrics are computed based on the testing set. In this study, all samples excluding those from baseline passes are used as the testing data. For example, if the baseline contains cuts 1-100, the testing set comprises 79 normal samples and 171 abnormal samples. An ideal TBD system should be able to identify the time when the abnormal condition appears, hence minimising FAR and maximising FDR.

Table 4.3 shows the results obtained by all methods. The fixed threshold, floating threshold, and OCSVM have almost 100% FDR values but also high FARs. The DBP is not accurate unless the baseline passes cover cuts that are close to RBP, which indicates that the threshold is too

Table 4.3: Tool breakage detection results; comparative study in the straight-line milling experiment

Method	BL	DBP	Δ	FDR(%)	FAR(%)	Acc(%)
The proposed	SDC-based	**179**	**1**	**95.3**	**25.0**	**94.9**
Fixed threshold	1-50	51	129	100	81.4	65.0
	1-100	102	78	100	84.8	73.2
	1-175	**181**	**1**	99.4	0.0	99.4
Floating threshold	Mic	122	58	100	33.0	83.3
	VibX	**181**	**1**	99.4	0.0	99.7
	VibY	150	30	100	17.0	91.4
	VibZ	54	126	100	71.6	63.7
OCSVM	1-50	51	129	100	99.2	57.3
	1-100	116	64	100	75.9	76.0
	1-175	**176**	**4**	100	100	97.7
ECOD	1-50	147	33	25.7	0.8	57.3
	1-100	310	130	9.4	0.0	38.0
	1-175	302	122	20.5	0.0	22.3

conservative to overcome subtle disturbance. While the ECOD method tends to trigger fewer alarms. This method is not sensitive to real anomalies and misses the tool breakage point. By contrast, the proposed method can detect RBP accurately and achieve a good trade-off between FDR and FAR.

4.4 Case 2: milling process with a circular tool path

4.4.1 Experimental setup

In this experiment, the dynamic milling strategy was adopted. The machine tool was DMG MORI's NVX5080. A cylinder TC4 workpiece was machined by 4 OSG's UVX-Ti-5F 16 mm-diameter 5-flute end mills with variable helix and unequal spacing. Centric circles from outer to inner comprise the tool path as shown in Figure 4.10 (a). The cutting parameters are listed

Table 4.4: Cutting conditions in the circular milling experiment

Tool	Spindle speed (RPM)	Feedrate (mm/min)	ADoC (mm)	RDoC (mm)	# of circles
1	2586	1448	19.5	2.4	29
2	2586	1448	19.5	2.4	29
3	2387	1002	19.5	2.4	24
4	2785	1560	19.5	2.4	43

in Table 4.4. Three sets of spindle speed and feed rate were considered, aiming to introduce process variability in the experiment. During machining, the forces on the plate and vibrations on the plate and spindle were sampled at 10240 Hz. The specifications of these sensors have been given in Table 3.1. The number of data channels is 9 in total.

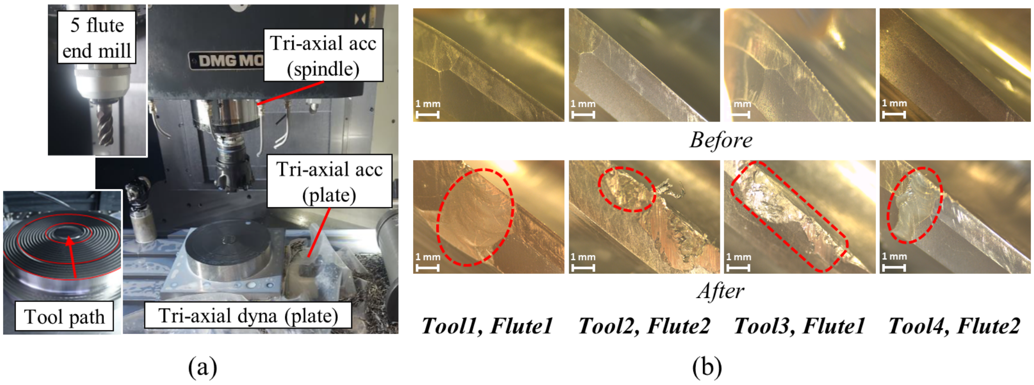


Figure 4.10: (a) Experimental setup and the tool path; (b) images of flutes before and after machining.

The tool condition was checked three times after outer, middle, and inner cuts, sequentially. In the test, the tools were used to machine only one or two layers of the workpiece, after which all tools were severely damaged. As shown in Figure 4.10 (b), unexpected chipping happens on some of the flutes at the end of the tools' lifecycle.

4.4.2 Experimental results

The data decomposition and analysis are similar to that in case 1. In this case, TDC1 and SDC1 are selected as the dominant components as well. The volume of data to be analysed is compressed from 9 sets of sensor signals to a single time series and a 9-dimensional vector. An AR(6) model-based PSD estimate was used to reveal the TPF-related dynamics. PSD amplitudes over 15 frequencies around TPF constituted the TDC1 feature vector. The

SDC1s were 9-dimensional vectors, showing different patterns especially when the tools were severely broken, as can be seen in Figure 4.11.

Here we select the first 5 cuts as initial passes. Table 4.5 shows the TBD results using the proposed and four existing methods. As the number of cuts is limited for the calculation of FDR and FAR, we evaluate the effectiveness by looking at if the DBP falls into the range of RBP. Firstly, only the proposed method and ECOD with a baseline of the first 20 passes show reasonable detection results. However, there is a risk of missing RBP for the latter as cut 20 is very close to the breakage. Besides, the fixed threshold and OCSVM methods determine cuts that are 1 or 2 passes following the BL as anomalies in all cases. These methods can not provide meaningful performance in practice. For the floating threshold-based method, the choice of feature has an effect on the accuracy. It is hard to determine such a feature from a single sensor signal so that the TBD results are acceptable for all cutters.

The rate that the integration of TDCs and SDCs can make a reliable judgement of tool breakage events. It is worthwhile to mention that an appropriate selection of l_{init} relies on the empirical estimation of the tool's lifecycle. The suggestion is to use 10% as the initialisation condition.

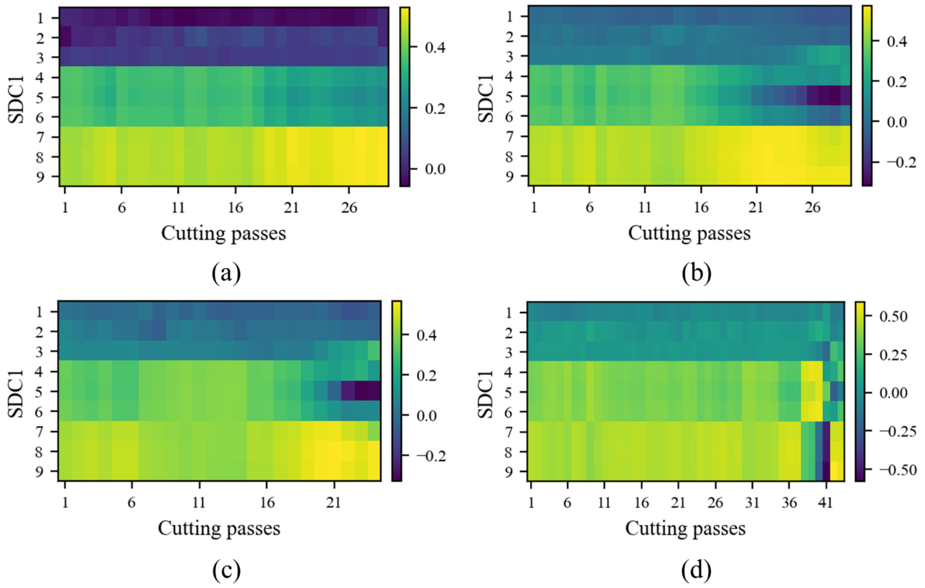


Figure 4.11: Heatmap of SDC1s vs. cutting passes: (a) Tool 1, (b) Tool 2, (c) Tool 3, (4) Tool 4.

Table 4.5: Tool breakage detection results; comparison study in the circular milling experiment

Method	BL	Tool 1	Tool 2	Tool 3	Tool 4
Real breakage point		15-29	15-29	15-24	38-43
The proposed	SDC-based	**28**	**17**	**23**	**37**
Fixed threshold	1-10	11	11	11	11
	1-15	16	16	16	26
	1-20	22	21	21	21
Floating threshold	ForceP	-	21	13	8
	VibP	9	7	11	8
	VibS	**29**	9	**15**	24
OCSVM	1-10	12	11	11	11
	1-15	16	16	16	16
	1-20	21	21	21	21
ECOD	1-10	12	14	21	16
	1-15	29	26	21	16
	1-20	**29**	**26**	**23**	**37**

4.5 Discussion

In this study, the novel concept of TSD has been applied to perform unsupervised detection of tool breakage. Its effectiveness has been demonstrated in milling experiments. In order to provide potential users with the applicability of the proposed method, some discussions are summarised as follows.

1. *The target systems:* the proposed method has potential applications in anomaly detection for various industrial processes (e.g., rotor system, chemical production) that face similar failure risks like tool breakage. As components undergo wear and tear, their performance may degrade gradually. However, sudden events such as component failure result in step-change anomalies. Conventional methods, relying on a fixed baseline, struggle to address this issue as any deviation from the baseline is incorrectly identified as an anomaly. The proposed method uses sensor domain information to establish and update the baseline condition, accommodating normal variability while

effectively capturing step changes by continuous threshold updates.

2. *The multi-sensor system:* the multi-sensor system lays the foundation of the proposed method. Many aspects should be considered carefully including the budget, sensor integration and synchronisation problem. Firstly, The multi-sensor system can be expensive, in terms of both purchase cost and maintenance. A feasible solution should be economically affordable. Secondly, careful checking and pre-processing of the data are necessary such that different types of sensors can be integrated effectively. For example, the AE sensor has a much higher frequency range (50-400 kHz) compared with the accelerometer (0-5 kHz), effective information does not remain after down-sampling. Besides, data synchronisation is also a challenge when a large number of sensors are used. In our experiment, the data acquisition module can synchronise measurements across a network. In a more sophisticated application, advanced synchronisation algorithms should be developed. In principle, provided these issues have been addressed, the proposed method can work in a complicated multi-sensor system.

3. *The interpretation of TDC/SDC:* the TDCs can be interpreted as a linear combination of original signals with the coefficients determined by the SDCs. Many studies only pay attention to the TDCs for the purpose of dimensional reduction and claim that it is hard to explain the meaning of SDCs. The present study reveals, for the first time, the significance of SDCs and shows how SDCs can be used to indicate the gradual changes in dynamics of a milling process. In Figure 4.7 (a), it can be seen that the first element of SDC1 has an increasing trend while the third element changes to negative in later stages. It is reasonable because the sound, which corresponds to the first element, would be louder after the happening of tool breakage. But the physical interpretation of the alteration of the third element, which is the y-axial vibration signal, still needs clarification. Besides, it is expected that the SDC would change continuously and smoothly in a gradual tool wear process. If so, the relationship between the pattern of SDCs and tool wear could be found and used for tool wear monitoring.

4. *Feasibility study:* it is suggested to conduct pilot testing or proof-of-concept experiments to assess the effectiveness of the anomaly detection algorithm in a controlled environment. Through the feasibility analysis, the users will establish an initial understanding about how the TDCs and SDCs change in their problems. Next, fine tuning of key parameters such as the number of initial passes and the characteristic features of TDC series can be conducted to optimise the accuracy and reliability of the algorithm.

5. *Limitations:* it is worthwhile mentioning that in some scenarios, the algorithm may not

be effective. For instance, when the anomaly-induced changes are too subtle to be identified by the threshold, the adaptive baseline may include abnormal samples and lead to missed alarms. In this case, updating the threshold more frequently can improve its sensitivity to subtle anomalies, which will be the main focus of next chapter.

4.6 Conclusion

This chapter proposes a novel process anomaly detection approach based on the proposed time-sensor domain data decomposition and analysis framework. The novelty lies in the synthesis of time and sensor domain information. After the data decomposition, representative features are extracted from the TDCs. Meanwhile, a sequential clustering algorithm is applied to analyse SDCs to obtain valuable information associated with process conditions. In this way, the baseline stages and corresponding threshold can be determined and updated adaptively. The Mahalanobis distance-based method is then proposed to perform the anomaly detection. The proposed approach has been applied to the analysis of sensor data from two practical machining experiments for tool breakage detection. Comparative studies with existing methods have also been conducted. The results have demonstrated the effectiveness, the potential in process anomaly detection and the advantages of the new approach over existing techniques in terms of both computation time and overall accuracy in anomaly detection.

Currently, the TSD is applied to each data block one by one. The obtained TDCs and SDCs may have unexplained changes due to environmental noises, for example in Figure 4.7 (a), the SDC1s obtained around cut 150 show a similar pattern as around the tool breakage point. In fact, most industrial processes are changing continuously. The time-dependent relationship among different data blocks should be considered when applying the TSD framework such that the process changes can be revealed clearly. In the next chapter, a recursive TSD algorithm will be proposed to realise time-varying process monitoring.

5

Time-varying process monitoring based on recursive time-sensor decomposition and application to tool wear monitoring

5.1 Introduction

In this chapter, a novel time-varying process monitoring approach is proposed. Instead of applying TSD to each data snapshot separately, the influence of previous snapshots on the current data is considered in a recursive way. In this sense, the TSD algorithm is further extended to a recursive TSD (RTSD) method. After the data decomposition, the time and sensor domain components are better related to the overall changes and underlying trend of the process, respectively, making them applicable to time-varying process monitoring.

As mentioned in Chapter 1, the data-driven process monitoring methods have currently become a research hotspot. One approach is referred to as multivariate statistical process monitoring (MSPM). MSPM operates by evaluating the statistical deviation of multivariate measurements from a predefined normal region. When the index of deviation exceeds a predetermined threshold, potential faults or abnormal conditions are detected. However, a limitation of traditional MSPM approaches is that the sensitivity to minor mean shifts decreases as the number of process variables increases [167]. Consequently, in practice, MSPM often employs dimension-reduction techniques to handle high-dimensional data. Techniques such as principal component analysis (PCA), partial least squares (PLS), and their extensions are commonly used to address dimension reduction problems [168, 169].

A basic assumption behind MSPM approaches using PCA or PLS is that the monitoring variables are independent and identically distributed. Ideally, the process states should be stationary such that the estimated mean and covariance can fully describe the process. Nonetheless, these foundational assumptions may be readily violated, as complex industrial processes are inherently time-varying [2, 3]. Factors such as catalyst deactivation, equipment degradation, and fluctuating operating conditions contribute to the time-varying change [170]. Consequently, measurements may be nonstationary exhibiting deterministic but unknown changes in mean and variance. For example, the milling process in manufacturing is a time-varying process where the sensor data display a deterministic but unknown trend of change due to gradual increases in tool wear [171]. Fig. 5.1 shows an illustration of the data distribution of stationary and time-varying processes. It can be found that in a time-varying process, the faulty condition cannot be detected easily by a conventional MSPM approach, in which a fixed control limit (CL) is adopted and normal time-varying behaviours may trigger false alarms. To address this issue, a monitoring scheme should be updated to accommodate time-varying behaviours while still able to detect abnormal conditions [172]. For example, the adaptive CL shown in Fig. 5.1 can be a solution which follows the gradual change of process variables and trigger alarms only when an abrupt change in these variables happens. According to the literature, this adaption can be achieved at either the data modelling stage or the control chart creating stage. The former relies on eliminating data variations induced by the time-varying behaviour of a process and only focusing on the stationary behaviours of the process, while the latter requires designing control charts such that normal process changes are incorporated into the in-control states.

When adaptation is realised at the data modelling stage, methods based on adaptive PCA/PLS [169, 173], slow feature analysis (SFA) [174, 175] and cointegration analysis (CA) [176, 177] are commonly employed. The adaptive MSPM approaches handle the time-varying property by concurrently updating the process model using both current and previous data, based on a moving window or recursive strategy. In the moving window method, a new model is generated from the latest samples within a fixed-length window. Comparatively, the recursive strategy utilises all preceding samples with decaying weights for model updates. For example, Li *et al.* [169] proposed a recursive PCA (RPCA) method for the monitoring of time-varying processes with slow and normal changes. Grasso *et al.* [178] utilised moving window

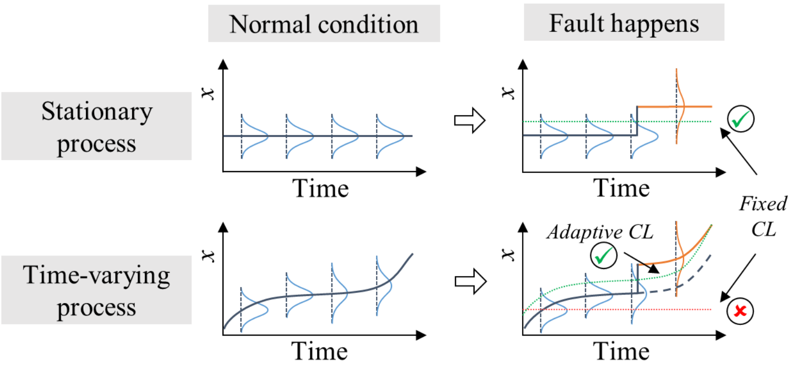


Figure 5.1: An illustration of the data distribution of stationary and time-varying processes and the fixed and adaptive control limits.

PCA (MWPCA) and adaptive control limit (CL) to detect tool breakage. SFA is an unsupervised learning algorithm that is used to extract slowly varying trends from a fast-fluctuating input signal. This makes it useful in the monitoring of time-varying processes, especially with complex evolving trends. In [175], SFA was utilised to monitor multi-phase batch processes by building different models for local steady states and global dynamics, respectively. CA is a statistical tool for determining the long-run equilibrium relationship between multiple nonstationary time series. In process monitoring, different variables may share the same long-run trend due to the time-varying behaviour. This long-term relationship is used by CA to characterise an anomaly or shift in process operational conditions [177].

When adaptation is implemented via control chart creating, researchers have proposed adaptive methods that continuously update the monitoring samples [179, 180] or control limits [181, 182]. Typically, the previous samples are directly used to generate a new vector by either a moving window or a recursive updating technique [180]. Monitoring based on this new vector can provide more sensitivity to minor and moderate shifts in the mean vector of time-varying processes. This method creates the so-called multivariate cumulative sum (MCUSUM) [183] or exponentially weighted moving average (MEWMA) control charts [184]. On the other hand, the adjustment of the control limit is usually exploited in adaptive MSPM schemes. In RPCA, the control limits are adjusted according to the number of principal components (PCs), which might change due to the model updates. Wang *et al.* adjusted the control limit according to the distribution of previous control limits [181]. However, in some cases, where the faults are gradual and subtle, the updated control limit may adapt to the faults as well, thus failing to trigger alarms. To resolve this issue, Wang *et al.* introduced an N-step-ahead prediction method utilising the model obtained in the Nth step ahead to calculate monitoring statistics and control limits, thereby enhancing the sensitivity of the control chart to ramp faults [182]. Most of the control charts only contain phases I and II corresponding to in-control and undetermined states, respectively, which might not work for time-varying processes. Recently, researchers have improved the design of two-phase control charts and proposed a novel three-region control chart [185]. This new design incorporates an additional region between the worst acceptable and the best unacceptable conditions, thereby facilitating the assessment of states associated with gradual system degradation.

However, the existing methods still face significant challenges in practical implementation. Firstly, the sample-wise monitoring scheme suffers from a huge computational burden and a high sensitivity to external noises [186]. Secondly, some theoretical assumptions are not applicable in practical scenarios, leading to failure in accommodating complex and unknown trends of variations [187]. Moreover, to ensure timely fault detection in time-varying processes, key parameters such as the moving window length and the number of prior steps need to be tuned carefully. There is a lack of an optimisation algorithm working for different processes.

The present study aims to resolve these challenges by a novel RTSD algorithm, which directly utilises the correlation structure of the measurements and avoids making unrealistic assumptions about the data or the process itself. In Chapter 4, SDCs are used to reflect the health status of processes and to determine a reasonable baseline in the tool breakage detection system. In RTSD, the data correlation matrix is updated recursively. The generated TDCs and SDCs present different time-varying properties of a process under study. Then a novel control chart with adaptive control limits, combining the time and sensor domain information, is designed. This novel scheme is able to adapt to normal process changes and trigger alarms only when significant faults take place. Applications in a simulation study and a tool wear process have demonstrated that this novel framework can adapt to normal but slow variations with degradation of equipment or process and timely detect faults and abnormal conditions, without a need for prior knowledge of underlying processes.

The main contribution of this Chapter is the innovative RTSD algorithm which, for the first time, exploits SDC and TDC to represent the general trend and the overall time-varying behaviours (including both normal and fault-induced variations) of an industrial process, respectively. This enables the creation of novel TDC-based control chart statistics, and SDC-based control limits, ensuring a good trade-off between fault detection rate and false alarm rate and making the proposed approach applicable to time-varying industrial process monitoring.

5.2 Problem description

5.2.1 Data structure in block-wise MSPM

In this study, the process monitoring is executed in a block-wise manner, where a monitoring result is generated after collecting a block of multi-sensor data. Compared with the sample-wise monitoring, this strategy is more efficient and reasonable, especially considering the relatively high sampling rate for process changes. The selection of block size depends on specific problems, for example, a continuous milling process. The underlying assumption is that the process condition remains unchanging over a certain phase of a data block. It should be noted that the proposed framework can be adapted to a sample-wise monitoring scenario if necessary.

The raw data for the entire process can be organised as a three-dimensional (3D) array as in the case of batch processes monitoring [188]. It is worth mentioning that in batch process monitoring, the data are repeated batch-wise, where each batch may contain multiple phases corresponding to different production operations. On the other hand, the focus of this study is on handling successive data across a single production process. Fig. 5.2 (a) illustrates the structure of the 3D array which consists of K block matrices stacked phase-wise. Each matrix contains information about process conditions over each phase. Each phase here may represent a fixed sampling period or a physical period of process operations.

To relieve the computation burden, it is necessary to decompose the array into a two-dimensional matrix. Among various types of decomposition, phase-unfolding is widely used. As shown in Fig. 5.2 (a), the dimension of phase is retained while the other two dimensions are merged to form a new vector. In practice, different forms of the 3D array have been considered, leading to $\underline{\mathbf{X}}$ with different sizes. These forms of 3D array include,

1. The multi-sensor data, which is denoted by $\underline{\mathbf{X}} \in \mathbb{R}^{I \times J \times K}$. As shown in Fig. 5.2 (b), I, J, and K are the number of samples, variables, and phases, respectively.

2. The dimensional-reduction data, which has the form $\underline{\mathbf{X}} \in \mathbb{R}^{I' \times J' \times K}$. As denoted in Fig. 5.2 (c), I' and J' represent the number of principal components, along the original I and J-directions, obtained through multi-way dimension reduction algorithms.

3. The data features, which are organised as $\underline{\mathbf{X}} \in \mathbb{R}^{P \times J \times K}$ corresponding to Fig. 5.2 (d). In this case, P denotes the number of features extracted from each variable.

4. When the number of samples equals one, the raw data degrades to a two-dimensional matrix $\mathbf{X} \in \mathbb{R}^{J \times K}$ naturally, which corresponds to the sample-wise monitoring problem.

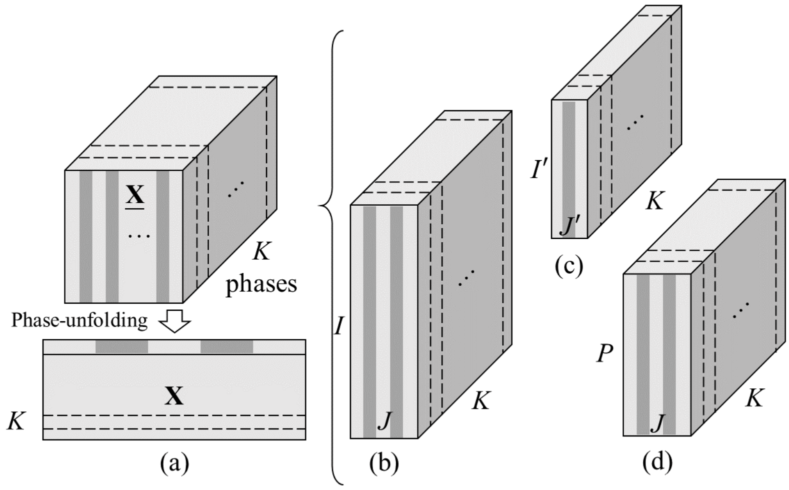


Figure 5.2: (a) Phase-unfolding of the three-dimensional array; illustration of (b) raw data; (c) dimensional-reduction data; (d) data features.

In block-wise MSPM, it is assumed that the process condition keeps unchanging over the time period of one block. The general MSPM framework can be implemented on matrix $\mathbf{X}$ after data unfolding. Faulty states are detected when a new phase of samples is significantly different from the samples in normal phases. In time-varying industrial processes, the statistical nature of the measurements such as mean and correlation may be different over different phases, making it difficult to identify faults in the processes.

5.2.2 Monitoring strategy

A monitoring procedure normally includes dimension reduction (if necessary), monitoring model parameter estimation and control limit determination (Phase I), and process state monitoring (Phase II). Let K_1 denote the number of the phases of normal samples used in Phase I. The mean vector and covariance matrix are used as monitoring model parameters. The quality indices can be calculated to represent process conditions. In this section, two multi-way PCA approaches including vectorised PCA (VPCA) and multi-linear PCA (MPCA) [105, 189] will be introduced first. Then three time-varying process monitoring approaches including adaptive PCA, SFA and CA [169, 175, 177] are revisited briefly.

Vectorised PCA: In this method, the dimension of the unfolded matrix $\mathbf{X}$ is $IJ \times K$. Each column consists of data from all samples and variables within one block. The mean and covariance matrix are estimated by $\boldsymbol{\mu} = \frac{1}{K_1}\sum_{k=1}^{K_1} x_k$ and $\boldsymbol{\Sigma} = \frac{1}{K_1-1}\sum_{k=1}^{K_1}(x_k - \boldsymbol{\mu})(x_k - \boldsymbol{\mu})^T$.

Assume $\mathbf{U} \in \mathbb{R}^{IJ\times r}$ consists of the eigenvectors of $\mathbf{\Sigma}$ corresponding to its first r largest eigenvalues $\lambda_1, \ldots, \lambda_r$, the principal component of a sample $\boldsymbol{x}$ is given by $\boldsymbol{z} = \mathbf{U}^T(\boldsymbol{x} - \boldsymbol{\mu})$. Two monitoring statistics, Hotelling's T^2 and Q-statistic can be calculated by

$$T^2 = \boldsymbol{z}^T \mathbf{\Lambda}^{-1} \boldsymbol{z} \tag{5.1}$$

$$Q = (\boldsymbol{x} - \boldsymbol{\mu})^T (\mathbf{I} - \mathbf{U}\mathbf{U}^T)(\boldsymbol{x} - \boldsymbol{\mu}) \tag{5.2}$$

where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, ..., \lambda_r)$.

Under the assumption of multivariate normality of the measurements and temporal independence, the control limit can be determined theoretically. For example, F-distribution is usually used to estimate the control limit for Hotelling's T^2 control chart by

$$\begin{aligned} \mathrm{UCL} &= \frac{r(K_1+1)(K_1-1)}{K_1^2 - K_1 r} F_{\alpha;r,K_1-r} \\ \mathrm{LCL} &= 0 \end{aligned} \tag{5.3}$$

where $F_{\alpha;r,K_1-r}$ is the upper α percentage point of a F-distribution with parameters r and $K_1 - r$. UCL and LCL represent upper and lower control limit, respectively. The control limit corresponding to the upper α percentile of the Q-statistic can be calculated by

$$\mathrm{UCL} = \theta_1 \left(\frac{z_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (1 - h_0)^2}{\theta_1^2} \right)^2 \tag{5.4}$$

where $\theta_i = \sum_{j=r+1}^{IJ} \lambda_j^i$ for $i = 1, 2, 3$ and $h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2}$ and z_α is the upper α percentile of the standard normal distribution. Note that if the number of observations is large, the control limits can be approximated using empirical distributions [3]. In the monitoring stage, if the statistics of a new block of samples exceed the range of control limits, the system will trigger off an alarm.

Multi-linear PCA: In this case, the dimensional-reduction data is used, which means $\mathbf{X} \in \mathbb{R}^{I'J'\times K}$. The mean and covariance matrix can be estimated in the same way as in VPCA. Then Hotelling's T^2 statistic in eq (5.1) becomes

$$T^2 = (\boldsymbol{x} - \boldsymbol{\mu})^T \mathbf{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})' \tag{5.5}$$

The Phase II control limit for this statistic has the same form as eq (5.3) with r replaced by $I'J'$. Note that both VPCA and MPCA methods assume that the mean and variable correlations do not change over normal phases. These can not adapt to normal process variations. When applied to the time-varying process, these approaches could be too sensitive to distinguish normal and faulty samples and trigger more false alarms [186].

Adaptive PCA: Both raw data and data features can be used in this method. Taking the feature matrix as an example, the matrix $\mathbf{X}$ has a size of $PJ \times K$. Different from VPCA and MPCA, the mean and covariance matrix in RPCA need to be updated every time a block of normal samples is available by

$$\boldsymbol{\mu}^k = (1-\eta)\boldsymbol{x}^k + \eta\boldsymbol{\mu}^{k-1} \tag{5.6}$$

and

$$\boldsymbol{\Sigma}^k = (1-\eta)(\boldsymbol{x}^k - \boldsymbol{\mu}^k)^T(\boldsymbol{x}^k - \boldsymbol{\mu}^k) + \eta\boldsymbol{\Sigma}^{k-1} \tag{5.7}$$

where η is the forgetting factor. Let $\mathbf{U}^k$ be the loading matrix. The principal component of $\boldsymbol{x}^{k+1}$ is $\boldsymbol{z}^{k+1} = \mathbf{U}^{k^T}(\boldsymbol{x}^{k+1} - \boldsymbol{\mu}^k)$. Now the quality statistics become

$$T^2_{k+1} = \boldsymbol{z}^{k+1^T}\boldsymbol{\Lambda}^{k^{-1}}\boldsymbol{z}^{k+1} \tag{5.8}$$

$$Q_{k+1} = (\boldsymbol{x}^{k+1} - \boldsymbol{\mu}^k)^T(\mathbf{I} - \mathbf{U}^k\mathbf{U}^{k^T})(\boldsymbol{x}^{k+1} - \boldsymbol{\mu}^k) \tag{5.9}$$

where $\boldsymbol{\Lambda}^k = \mathrm{diag}(\lambda_1, \ldots, \lambda_{r^k})$ and r^k is the number of principal components.

As the covariance matrix is always updated to incorporate new and historical information, the number of principal components and eigenvalues may change and thus the control limit is adjusted. The upper control limit for Hotelling's T^2 statistic and the Q-statistic as shown in (5.3) and (5.4) should change with k. This approach has the potential to adapt to slow changes and signal an abrupt shift [178], while the existence of outliers may lead to inaccurate estimation of the principal components and cause false alarms.

SFA and CA: Another way to deal with time-varying processes is to separate the general trend from stationary variation using time correlation analysis. SFA aims to extract slowly varying latent variables from time series. The data can be divided into long-term general trends and short-term fluctuations. CA yields a long-run dynamic equilibrium relationship between process variables. The residual sequence describes the dynamic equilibrium errors, which are stochastic and stationary. In both methods, when the normal and slow change is excluded, the remaining normal phases of samples should follow a multivariate normal distribution. The general MSPM framework is applied to monitor the condition of a new block of samples.

As can be seen, most existing methods try to extract stationary behaviours of an underlying process by excluding the long-term trend of variations from raw data. Then the control chart is created only using the statistics of the extracted stationary behaviours. An issue with this strategy is that the assumption about the potential trend of variations does not always hold, leading to inaccurate results. Therefore, there is a real need to develop a monitoring method that adapts to the trend of process variations automatically and triggers alarms only when a true fault occurs. In addition, this method should also be robust to outliers and easy to implement.

5.3 Methodology

The flowchart of the proposed novel MSPM approach for time-varying process monitoring is shown in Fig. 5.3. The implementation procedure has two phases. In Phase I, all available in-control samples are utilised to estimate the mean vector and covariance matrix. Meanwhile, the parameters used for control limit adjustment are also determined. In Phase II, both T^2 statistics and corresponding CL are calculated in real time when each block of new samples is collected and used for process monitoring. An alarm will be triggered if the T^2 statistic exceeds the CL. The novelty is associated with the RTSD-based calculation of T^2 statistic and CL. The RTSD algorithm decomposes raw data into the time domain components known as TDCs and the sensor domain components known as SDCs. The feature vectors extracted from the TDCs are used to calculate the T^2 statistic to represent the overall process changes. The CL is determined based on the distribution of SDCs to allow an adaptive update of CL and accommodate the normal trend of process variations. Details of this approach are introduced in the following subsections.

5.3.1 Recursive time-sensor decomposition

Consider the original measurement matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, where m and n denote the number of samples and sensors, respectively. For the kth block, we have the data matrix $\mathbf{X}^k = [\boldsymbol{x}_1^k, \boldsymbol{x}_2^k, \ldots, \boldsymbol{x}_n^k]$, and the length of $\boldsymbol{x}_j^k$ is m^k. Firstly, the raw data matrix is normalised as follows:

$$\bar{x}_{i,j}^k = (x_{i,j}^k - \mu_j^1)/\sigma_j^1, \quad i = 1, 2, \ldots m^k, j = 1, 2, \ldots, n \tag{5.10}$$

where μ_j^1 and σ_j^1 denote the mean and standard deviation of the jth sensor signal in the first block. This step ensures that the data from different sensors are in the same range. The first and second-order statistics of $\mathbf{X}^1$ are selected as the reference such that the relative difference between data statistics of the first and kth blocks is kept. Then, the correlation matrix of $\bar{\mathbf{X}}^k$ can be estimated as $\mathbf{R}^k = [r_{ij}]_{n \times n}$, where $r_{ij} = \mathrm{corr}(\bar{\boldsymbol{x}}_i^k, \bar{\boldsymbol{x}}_j^k)$ denotes the correlation coefficient between the ith and jth normalised signals. In this study, the recursively updating strategy

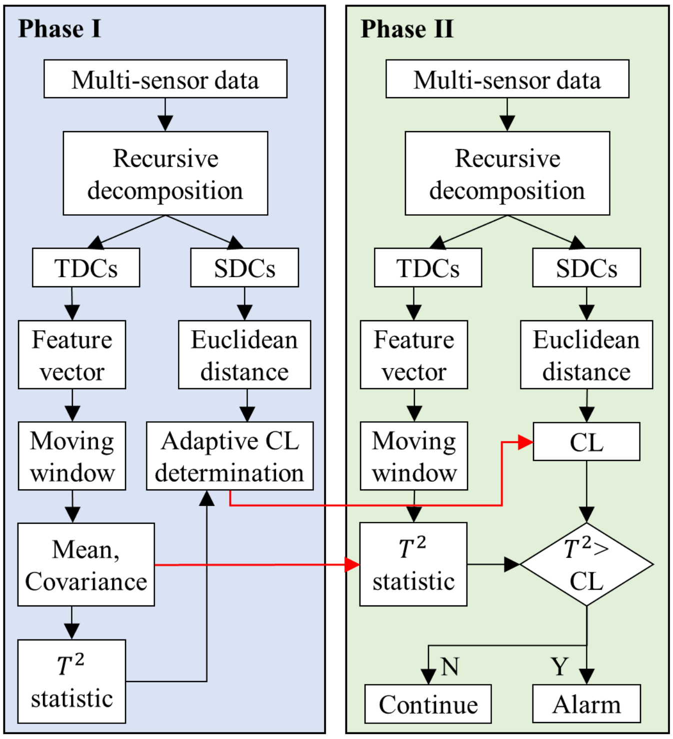


Figure 5.3: Flowchart of the proposed process monitoring system based on RTSD.

is introduced to recalculate the correlation matrix as

$$\hat{\mathbf{R}}^k = \frac{\sum_{i=1}^{k} \eta^{k-i} \mathbf{R}^i}{\sum_{i=1}^{k} \eta^{k-i}} = \frac{1-\eta}{1-\eta^k} \mathbf{R}^k + \eta \frac{1-\eta^{k-1}}{1-\eta^k} \hat{\mathbf{R}}^{k-1} \approx (1-\eta)\mathbf{R}^k + \eta \hat{\mathbf{R}}^{k-1} \tag{5.11}$$

where $0 \leq \eta \leq 1$ denotes the forgetting factor. The approximation in equation (5.11) is reasonable for a sufficiently large k. Hence, $\hat{\mathbf{R}}^k$ is essentially a weighted mean of all available correlation matrices. The magnitude of the weight for $\mathbf{R}^i$ decreases exponentially as i decreases, i.e., the most recent samples have more influence on the estimation result than the older samples.

As $\hat{\mathbf{R}}^k$ is a positive definite matrix, it has an eigendecomposition $\hat{\mathbf{R}}^k = \mathbf{\Phi}^k \mathbf{\Lambda}^k \mathbf{\Phi}^{k^T}$, where $\mathbf{\Lambda}^k = \mathrm{diag}(\lambda_1^k, \lambda_2^k, \ \ldots, \lambda_n^k)$ is a diagonal matrix containing the eigenvalues in descending order, and $\mathbf{\Phi}^k = [\varphi_1^k, \varphi_2^k, \ldots, \varphi_n^k] \in \mathbb{R}^{n \times n}$ consists of the corresponding eigenvectors. These vectors are defined as SDCs. Then, the TDCs can be calculated by

$$\mathbf{Y}^k = \bar{\mathbf{X}}^k \mathbf{\Phi}^k = [y_1^k, y_2^k, \ldots, y_n^k] \in \mathbb{R}^{m^k \times n} \tag{5.12}$$

Noting that $\mathbf{\Phi}^k$ is a unitary matrix, the decomposition of $\bar{\mathbf{X}}^k$ holds as shown in

$$\bar{\mathbf{X}}^k = \mathbf{Y}^k \mathbf{\Phi}^{k^T} = \sum_{i=1}^{n} y_i^k \varphi_i^{k^T} \approx \sum_{i=1}^{n'} y_i^k \varphi_i^{k^T} \tag{5.13}$$

where n' denotes the number of dominant TDC/SDC pairs, which can be determined based on the magnitude of λ_i and cumulative percentage of variance (CPV) as introduced in Chapter 3. This study chooses the first TDC/SDC pair as the dominant one for simplicity.

The decomposition of the raw 3D array is illustrated in Fig. 5.4. Unlike existing methods, the raw data is decomposed into two 3D arrays in the time/sensor domain and then unfolded, respectively. This allows the information in multi-sensor signals to be analysed in two different domains, providing time-varying information of the overall process changes and the trend of process variations, respectively.

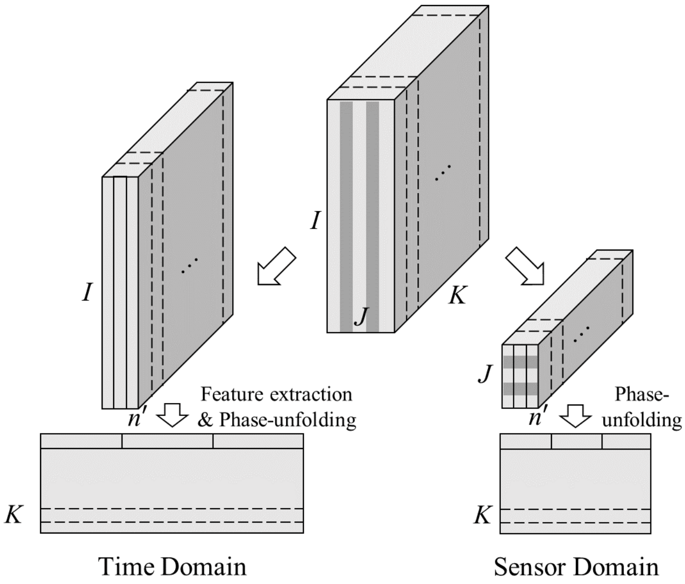


Figure 5.4: Illustration of the proposed recursive time-sensor decomposition of the 3D data matrix.

For the kth block, the time domain information of dominant TDCs can be represented by

$$\boldsymbol{f}^k = [f^k_{1,1}, f^k_{1,2}, \ldots, f^k_{1,p}, f^k_{2,1}, \ldots, f^k_{n',p}]^T \tag{5.14}$$

where $f^k_{i,j}$ denotes the jth feature extracted from the ith TDC $\boldsymbol{y}^k_i$, and p is the number of features extracted from any one TDC. In practice, the common statistics including mean, variance, and skewness can be calculated as, for example,

$$f^k_{1,1} = \mu(\boldsymbol{y}^k_1) = \frac{1}{m^k} \Sigma^{m^k}_{t=1} y^k_1(t) \tag{5.15}$$

$$f^k_{1,2} = \sigma(\boldsymbol{y}^k_1) = \frac{1}{m^k} \Sigma^{m^k}_{t=1}[y^k_1(t) - \mu(\boldsymbol{y}^k_1)]^2 \tag{5.16}$$

$$f^k_{1,3} = \gamma(\boldsymbol{y}^k_1) = \frac{1}{m^k \sigma(\boldsymbol{y}^k_1)^3} \Sigma^{m^k}_{t=1}[y^k_1(t) - \mu(\boldsymbol{y}^k_1)]^3 \tag{5.17}$$

5.3.2 Moving window control chart

As mentioned above, control charts are widely used for multivariate process monitoring and control. The T^2 control chart displays T^2 statistics and control limit versus process time. If the quality index exceeds the range of upper and lower CLs, the monitoring sample might significantly deviate from the stable or in-control state due to abnormalities and faults. Thus the corresponding process is out of control.

In this study, the process state is represented by the time-domain feature vector. To cope with environmental noises, the moving average is applied to construct monitoring vectors. Firstly, the mean of the feature vectors within a moving window is calculated as

$$z_k = \begin{cases} \dfrac{1}{k}\displaystyle\sum_{t=1}^{k} \boldsymbol{f}^i, & 1 \leq k < w \\ \dfrac{1}{w}\displaystyle\sum_{t=k-w+1}^{k} \boldsymbol{f}^i, & k \geq w \end{cases} \tag{5.18}$$

where w denotes the window length. z_k is defined as the monitoring vector from the kth block of data. The corresponding covariance matrix of the feature vectors within the window is given by

$$\mathbf{S}_k = \begin{cases} \dfrac{1}{k}\displaystyle\sum_{i=1}^{k} (\boldsymbol{f}^i - z_k)(\boldsymbol{f}^i - z_k)^T, & 1 \leq k < w \\ \dfrac{1}{w}\displaystyle\sum_{i=k-w+1}^{k} (\boldsymbol{f}^i - z_k)(\boldsymbol{f}^i - z_k)^T, & k \geq w \end{cases} \tag{5.19}$$

Assume the first K_1 data blocks are selected as Phase I measurements, the T^2 statistic of z_k is given by

$$T_k^2 = n'p(z_k - \bar{z})^T \bar{\mathbf{S}}^{-1}(z_k - \bar{z}) \tag{5.20}$$

where $\bar{z} = \frac{1}{K_1}\sum_{k=1}^{K_1} z_k$ and $\bar{\mathbf{S}} = \frac{1}{K_1}\sum_{k=1}^{K_1} \mathbf{S}_k$ denote the mean of in-control samples and mean of in-control sample covariance matrices, respectively.

5.3.3 Control limit determination

The determination of the control limit has an essential influence on the performance of control charts. The previous section has pointed out that a fixed control limit is not applicable to time-varying processes due to unknown process trends. This study proposes a novel threshold adjustment strategy based on the distribution of SDCs. The basic idea is to extract the trend of process variations from SDCs and then use the result to adjust the corresponding control limit.

Let $\boldsymbol{\varphi}^k$ denotes the sensor domain vector that is formed by concatenating the dominant SDCs of the kth block, i.e., $\boldsymbol{\varphi}^{k^T} = [\boldsymbol{\varphi}_1^{k^T}, \boldsymbol{\varphi}_2^{k^T}, \dots, \boldsymbol{\varphi}_{n'}^{k^{\,T}}]$. Similarly, the sensor domain vectors in the first K_1 blocks are used as a reference. The Euclidean distance from $\boldsymbol{\varphi}^k$ to the reference centre is calculated by

$$D_k = (\boldsymbol{\varphi}^k - \bar{\boldsymbol{\varphi}})^T(\boldsymbol{\varphi}^k - \bar{\boldsymbol{\varphi}}) \tag{5.21}$$

where $\bar{\boldsymbol{\varphi}} = \frac{1}{K_1}\sum_{k=1}^{K_1} \boldsymbol{\varphi}^k$ is the mean of reference samples. Next, the obtained distances of all the $\boldsymbol{\varphi}^k$ in Phase I consist of a set $\tau_D = \{D_1, D_2, \dots, D_{K_1}\}$.

Let $\mathrm{mean}(\tau_D)$ and $\mathrm{std}(\tau_D)$ represent the mean and standard deviation of the set τ_D, respectively. The deviation coefficient c_k can be given by

$$c_k = |D_k - \mathrm{mean}(\tau_D)|/\mathrm{std}(\tau_D) \tag{5.22}$$

The control limits are determined based on the T^2 statistics in Phase I and the coefficient c_k. The formulations of UCL and LCL are given as follows,

$$\begin{aligned} \mathrm{UCL}_k &= \mathrm{mean}(\tau_{T^2}) + c_k \times \mathrm{std}(\tau_{T^2}) \\ \mathrm{LCL}_k &= \mathrm{mean}(\tau_{T^2}) - c_k \times \mathrm{std}(\tau_{T^2}) \end{aligned} \tag{5.23}$$

where $\tau_{T^2} = \{T_1^2, T_2^2, \dots, T_{K_1}^2\}$ is a set of all Phase I quality statistics. It can be seen that c_k defines the distance of UCL_k and LCL_k from $\mathrm{mean}(\tau_{T^2})$ in the sense of standard deviation. In a time-varying process, it is expected that the coefficient c_k can adapt to the unknown and deterministic trend of process variations. Thus adaptive control limits are produced.

To ensure the control limit can work for a specific process, two additional parameters α and β are introduced to modify UCL_k and LCL_k as $\mathrm{UCL}_k' = \alpha\mathrm{UCL}_k$ and $\mathrm{LCL}_k' = \beta\mathrm{LCL}_k$, which meet

$$\begin{aligned} \sum_{k=1}^{K_1} I(\mathrm{UCL}_k' \geq T_k^2) \geq 90\% K_1 \\ \sum_{k=1}^{K_1} I(\mathrm{LCL}_k' \leq T_k^2) \geq 90\% K_1 \end{aligned} \tag{5.24}$$

where $I(\cdot)$ is the indicator function. The control limit determination procedure is presented in **Procedure 1**. In Phase II, the statistic T_k^2 is compared with UCL_k' and LCL_k' for process condition monitoring.

Procedure 1 Control limit determination

```
1. Input: $\tau_{T^2} = \{T_1^2, T_2^2, \ldots, T_{K_1}^2\}$, $\tau_D = \{D_1, D_2, \ldots, D_{K_1}\}$, $\bar{\boldsymbol{\varphi}}$, $\alpha$ and $\beta$; the multi-sensor data
   $\mathbf{X}^k,\ k = K_1+1, K_1+2, \ldots, K$
2. Output: adaptive control limit $\mathrm{UCL}_k'$ and $\mathrm{LCL}_k'$
3. Obtain $\bar{\mathbf{X}}^k$ from $\mathbf{X}^k$ by Eq. (5.10)
4. Recursively update the correlation matrix $\hat{\mathbf{R}}^k$ by Eq. (5.11)
5. Decompose $\bar{\mathbf{X}}^k$ into TDCs $\boldsymbol{y}_i^k$ and SDCs $\boldsymbol{\varphi}_i^k$ by Eqs. (5.12) and (5.13), where $i = 1, 2, \ldots, n'$
6. Obtain the sensor domain vector $\boldsymbol{\varphi}^k$ by concatenating all $\boldsymbol{\varphi}_i^k$
7. Calculate the deviation coefficient $c_k$ by Eqs. (5.21) and (5.22)
8. Calculate $\mathrm{UCL}_k$ and $\mathrm{LCL}_k$ by Eq. (5.23)
9. Produce $\mathrm{UCL}_k' = \alpha\mathrm{UCL}_k$ and $\mathrm{LCL}_k' = \beta\mathrm{LCL}_k$
```

5.3.4 Relationship between SDCs and TDCs

As mentioned above, the sensor domain information is expected to reflect the trend of process variations due to the recursive updating of the correlation matrix. For the $(k-1)$th and kth blocks, the eigendecomposition of the correlation matrix is denoted as follows,

$$\begin{aligned}\hat{\mathbf{R}}^{k-1} &= \boldsymbol{\Phi}^{k-1}\boldsymbol{\Lambda}^{k-1}{\boldsymbol{\Phi}^{k-1}}^T \\ \hat{\mathbf{R}}^{k} &= \boldsymbol{\Phi}^{k}\boldsymbol{\Lambda}^{k}{\boldsymbol{\Phi}^{k}}^T\end{aligned} \tag{5.25}$$

Equation (5.11) gives the relationship between $\hat{\mathbf{R}}^{k-1}$ and $\hat{\mathbf{R}}^{k}$, i.e., $\hat{\mathbf{R}}^{k} \approx (1-\eta)\mathbf{R}^k + \eta\hat{\mathbf{R}}^{k-1}$.

Considering the forgetting factor η usually takes a value greater than 0.9, the matrix $\hat{\mathbf{R}}^{k-1}$ has primary influence upon the new correlation matrix $\hat{\mathbf{R}}^{k}$, indicating $\boldsymbol{\Lambda}^{k-1} \approx \boldsymbol{\Lambda}^{k}$ [190].

This implies either $\boldsymbol{\Phi}^{k} \approx \boldsymbol{\Phi}^{k-1}$ or $\boldsymbol{\Phi}^{k} \approx -\boldsymbol{\Phi}^{k-1}$ on the right hand of (5.25). In other words, $\boldsymbol{\varphi}_i^k \approx \boldsymbol{\varphi}_i^{k-1}$ or $\boldsymbol{\varphi}_i^k \approx -\boldsymbol{\varphi}_i^{k-1}$ holds for all $i = 1, 2, \ldots, n$. In theory, both situations are reasonable because both $-\boldsymbol{\varphi}_i^k$ and $\boldsymbol{\varphi}_i^k$ are eigenvectors of $\hat{\mathbf{R}}^{k}$. To eliminate this uncertainty, the implemented algorithm will choose $\boldsymbol{\varphi}_i^k$, which is close to $\boldsymbol{\varphi}_i^{k-1}$, as the ith SDC of the kth data block. As a result, in the n-dimensioanl space of SDC, the current SDC only deviates a little from the last SDC in terms of its direction. Therefore, the SDCs tend to remain stable and can reflect the general trend of process variations.

The TDCs are calculated as the linear combination of original measurements $\mathbf{Y}^k = \bar{\mathbf{X}}^k \mathbf{\Phi}^k$, indicating that all the changes in the data matrix, induced by both the normal trend and unexpected faults of a process, can be revealed by the TDCs. The extracted features of TDC can therefore be the overall process time-varying behaviours. This analysis implies that the proposed method can be effectively used for time-varying process monitoring.

5.3.5 Implementation algorithm

The proposed process monitoring algorithm consisting of Phase I and II is summarised in **Procedure 2**. In Phase I, the raw data is normalised to obtain $\bar{\mathbf{X}}^k$ by Eq. (5.10). Then the correlation matrix $\hat{\mathbf{R}}^k$ is updated recursively, from which the dominant TDCs and SDCs are generated by Eqs. (5.11)-(5.13). From TDCs, the feature vector $\boldsymbol{f}^k$ is extracted by Eqs. (5.14)-(5.17). The moving average strategy is applied to obtain the monitoring vector $\boldsymbol{z}_k$ as shown in Eqs. (5.18) and (5.19). Meanwhile, the dominant SDCs form the sensor domain vector $\boldsymbol{\varphi}^k$. After all the data in Phase I has been decomposed, the T^2 statistics and Euclidean distances are calculated respectively, forming the sets τ_{T^2} and τ_D. Then the parameters α and β are determined using the strategy introduced in Section III-C.

In Phase II, the RTSD is performed on each multi-sensor data block as in Phase I. Next, the T^2 statistic and adaptive control limit are calculated using Eqs. (5.20)-(5.23), respectively. If T_k^2 is within the range of LCL_k' and UCL_k', the process is still in control and continues. Otherwise, the process needs to be checked for any possible abnormal changes.

5.4 Case 1: simulation study

In this case, the performance of the proposed method is evaluated via Monte Carlo simulations. The multi-sensor data are generated from three benchmark signals proposed by Donoho and Johnstone [191], which are called 'blocks', 'heavysinve' and 'bumps', respectively. Many researchers have used these signals to test signal processing methods and statistical models [105, 192].

Procedure 2 **The proposed process monitoring algorithm**

```
1. Input: $\eta$, $K_1$, $w$, the multi-sensor data $\mathbf{X}^k,\ k = 1, 2, \ldots, K$
2. Output: whether the process state is in control or not
   In Phase I:
3. for $k = 1, 2, \ldots, K_1$
    3.1 Obtain $\bar{\mathbf{X}}^k$ from $\mathbf{X}^k$ by Eq. (5.10)
    3.2 Recursively update the correlation matrix $\hat{\mathbf{R}}^k$ by Eq. (5.11)
    3.3 Decompose $\bar{\mathbf{X}}^k$ into TDCs $\boldsymbol{y}_i^k$ and SDCs $\boldsymbol{\varphi}_i^k$ by Eqs. (5.12) and (5.13), where $i = 1, 2, \ldots, n'$
    3.4 Extract the time domain feature vector $\boldsymbol{f}^k$ by Eqs. (5.14)-(5.17)
    3.5 Obtain the monitoring vector $\boldsymbol{z}_k$ and covariance matrix $\mathbf{S}_k$ from $\boldsymbol{f}^k$ within a moving window by Eqs. (5.18) and (5.19)
    3.6 Obtain the sensor domain vector $\boldsymbol{\varphi}^k$ by concatenating all $\boldsymbol{\varphi}_i^k$
4. end for
5. Calculate $T^2$ statistics of $\boldsymbol{z}_k$ by Eq. (5.20), forming $\tau_{T^2}$
6. Calculate Euclidean distances of $\boldsymbol{\varphi}^k$ by Eq. (5.21), forming $\tau_D$
7. Determine the parameters $\alpha$ and $\beta$ by Eqs. (5.22)-(5.24)
   In Phase II:
8. for $k = K_1 + 1, K_1 + 2, \ldots, K$
    8.1 Run steps 3.1~3.6 to obtain $\boldsymbol{z}^k$ and $\boldsymbol{\varphi}^k$
    8.2 Calculate the statistic $T_k^2$ of $\boldsymbol{z}^k$ by Eq. (5.20)
    8.3 Calculate $\mathrm{UCL}_k'$ and $\mathrm{LCL}_k'$ based on $\boldsymbol{\varphi}^k$ by Eqs. (5.21)-(5.23)
    8.4 if $\mathrm{LCL}_k' \leq T_k^2 \leq \mathrm{UCL}_k'$
            Process continues
    8.5 else
            Stop and check if the state is really out of control
    8.6 end if
9. end for
```

5.4.1 Data generation

The benchmark signals denoted as $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, and $\boldsymbol{u}_3$ are shown in Fig. 5.5 (a), (b), and (c). The number of data points is 2000, i.e., $m = 2000$. Assume four variables are used for process monitoring ($n = 4$), the data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ can be generated from benchmark signals as follows,

$$\begin{aligned}
\boldsymbol{x}_1 &= b_1\boldsymbol{u}_1 + b_2\boldsymbol{u}_2 + \varepsilon_1 = \boldsymbol{s}_1 + \varepsilon_1 \\
\boldsymbol{x}_2 &= b_3\boldsymbol{u}_1^2 + b_4\boldsymbol{u}_3 + \varepsilon_2 = \boldsymbol{s}_2 + \varepsilon_2 \\
\boldsymbol{x}_3 &= b_5\boldsymbol{u}_2^2 + b_6\boldsymbol{u}_3^2 + \varepsilon_3 = \boldsymbol{s}_3 + \varepsilon_3 \\
\boldsymbol{x}_4 &= b_7\boldsymbol{u}_1\boldsymbol{u}_2 + \varepsilon_4 = \boldsymbol{s}_4 + \varepsilon_4
\end{aligned} \tag{5.26}$$

where ε_n is a random disturbance with $\varepsilon_n \sim \mathcal{N}(0, \sigma_{\varepsilon_n}), n = 1, \ldots, 4$, and $\boldsymbol{b} = [b_1, \ldots, b_7]^T$ is the model parameter vector, which is sampled from the multivariate normal distribution $\mathcal{MN}(\boldsymbol{\mu}_{\boldsymbol{b}}, \boldsymbol{\Sigma}_{\boldsymbol{b}})$. Similar to [105], the following settings were used in the simulation,

$$\begin{aligned}
\boldsymbol{\mu}_{\boldsymbol{b}} &= [0.2, 1, 1.5, 0.5, 1, 0.7, 0.8]^T \\
\boldsymbol{\Sigma}_{\boldsymbol{b}} &= \mathrm{diag}(0.08, 0.015, 0.05, 0.01, 0.09, 0.03, 0.06)
\end{aligned} \tag{5.27}$$

To represent the development of the whole process, 4000 blocks of data matrices are generated in total. Two types of process changes associated with the mean and variance of the measurements are defined as follows,

(i) Mean shift of the benchmark signals: $\boldsymbol{u}_i^k = \boldsymbol{u}_i + a_i^k \sigma_{\boldsymbol{u}_i},\ i = 1, 2, 3.$

(ii) Variance increase of the random noise: $\sigma_{\varepsilon_n}^k = d^k \sigma_{\boldsymbol{s}_n},\ n = 1, 2, 3, 4.$

where $k = 1, 2, \ldots, 4000$, denoting the kth block, and a_i^k and d^k are the process state-related coefficients.

The changes in coefficients a_i and d are shown in Figs. 5.5 (d) and (e), respectively. The time-varying behaviour is simulated by increasing the mean of benchmark signals and the power of noises. The normal evolution of a_i consists of a shorter but rapidly increasing stage and then a longer but gradually increasing stage as the blue line in Fig. 5.5 (d). This setting coincides with the degradation of lots of industrial equipment. Similarly, d following a normal evolution presents a slightly increasing trend as well. Two out-of-control situations are introduced from block 3001. The fault types are ramp and step changes as shown by the red lines in Figs. 5.5 (d) and (e). To be specific, under the faulty condition, either the mean of $\boldsymbol{u}_i$ or the variance of noise ε_n will deviate from the trend it should follow under normal conditions. When the fault happens on a_i, d follows the normal increasing pattern, and vice versa.

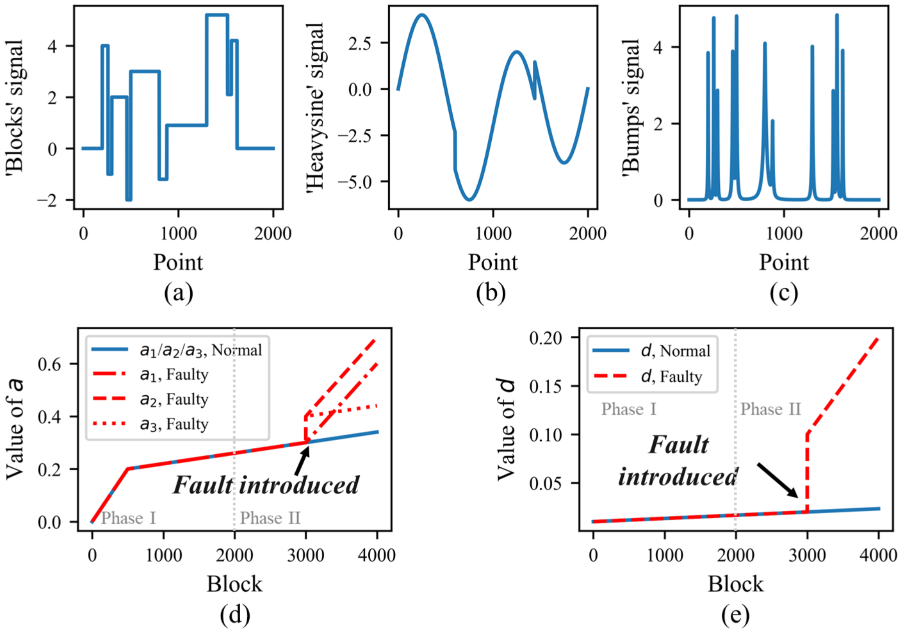


Figure 5.5: The benchmark signals: (a) 'Blocks', (b) 'Heavesine', and (c)'Bumps'; process state-related coefficients versus blocks: (d) coefficient a_i governing mean shift, (e) coefficient d governing variance increase.

Table 5.1: Monitoring results in the simulation study

Fault	Metrics	RTSD T^2	VPCA T^2	VPCA SSE	MPCA T^2	MPCA SSE	RPCA T^2	RPCA Q	SFA T_d^2	SFA T_e^2	CA T^2
a_i: faulty d: normal	FDR(%)	**99.1**	80.4	80.5	52.6	60.4	41.2	51.9	92.1	61.0	90.0
	FAR(%)	**1.1**	10.6	9.2	3.4	2.0	9.7	1.8	12.1	12.1	11.5
a_i: normal d: faulty	FDR(%)	98.7	21.5	23.8	42.9	6.9	60.8	**100.0**	**100.0**	99.4	**100.0**
	FAR(%)	**0.4**	10.3	10.7	2.9	2.1	11.1	4.6	18.3	13.1	16.6

5.4.2 Monitoring results

In this case, the first 2000 blocks of data are used in Phase I and the remaining in Phase II as shown in Figs. 5.5 (d) and (e). This means during the monitoring, the first half of data matrices are from normal processes and the second half from processes with fault. The proposed RTSD algorithm is applied to the simulation data following steps presented in **Procedure 2**. As a comparison, five methods based on VPCA [105], MPCA [105], RPCA [169], SFA [175] and CA are also implemented on the same data set [177].

In the proposed algorithm, the forgetting factor η is set to 0.999 and the length of the moving window w is 30. The extracted features include the variance, root mean square, and peak-to-peak value of TDC1. Only the upper control limit is considered because the out-of-control scenarios correspond to the increase of signal mean or variance. In VPCA and MPCA methods, the 3D array is processed directly. In Phase I, the array has the dimension of $2000 \times 4 \times 2000$. In VPCA, the array is unfolded first and then projected to the space of PCs. In MPCA, the dimensions of samples and variables are compressed, respectively. Then the new array is unfolded. The CPV used to determine the number of PCs is set to 95% in both methods. In Phase II, the data is projected to the same PCs space obtained in Phase I. The quality statistics including T^2 and SSE statistics are calculated in the same way as in [105].

For RPCA, SFA, and CA methods, the raw data cannot be used directly as the total number of samples is huge. Instead, the same three features are extracted from each variable in one block of data. Hence, the raw 3D array becomes a 12×4000 matrix. In RPCA method, the first 2000 samples are used for initialisation. The recursive model is updated for all the remaining samples. The forgetting factor is set to 0.999. The CPV of PCs is 95%. The T^2 and Q statistics are computed for process monitoring as shown in [169]. In SFA method, the first 2000 samples are utilised to generate a SFA model, which is applied to the samples in Phase II. The number of dominant features is determined according to [193]. Similarly, the monitoring statistics are calculated from the dominant slowest features (T_d^2) and the residuals (T_e^2), respectively. In CA method, the existence of cointegration between variables is checked first and then the cointegration rank is chosen through Johansen test [176]. A vector error-correction model is estimated to obtain the residual sequence, from which T^2 statistics are constructed.

In terms of the control limits, the adaptive limits for T^2 and Q statistics in RPCA are calculated as mentioned in [169], where the confidence level is set to 0.99. In all of the other methods used for comparison, the kernel density estimation (KDE) method is applied to fit the distribution of statistics of Phase I samples [194]. Then the control limit is determined with a confidence level of 0.99.

Table 5.1 presents the monitoring performance of these methods in terms of fault detection rate (FDR) and false alarm rate (FAR). In all scenarios, the proposed method can achieve almost 100% FDR and the lowest FAR. The T_d^2 of SFA and T^2 of CA also show high FDRs between 90% and 100%. But the FARs are above 10%, which means more normal samples are identified as faulty. The Q of RPCA works well in the second case but cannot detect fault in the first case accurately. Other statistics, including the T^2 and SSE of MPCA, T^2 of RPCA and T_d^2 of SFA, cannot achieve a good trade-off between FDR and FAR.

5.5 Case 2: dynamic milling experiment

To validate the performance of the proposed RTSD method in practice, a milling experiment has been conducted. The tool wear process is a typical time-varying process consisting of three stages: initial, gradual and rapid wear. The monitoring target is to detect the time when the tool becomes severely worn. This kind of fault usually accompanies with sudden and rapid increase in tool wear. The proposed RTSD is applied to analyse the collected multi-sensor data; other existing methods are also applied for comparative studies.

5.5.1 Experimental setup

The experiment was performed on a 5-axis milling machine DMG MORI's DMU 40evo. Three Sandvik CoroMill Plura 1630 solid carbide square shoulder 16 mm-diameter end mills with 4 flutes were used to machine a cylinder TC4 workpiece as shown in Fig. 5.6 (a). The dynamic milling strategy was adopted. The cutting parameters were set as follows: the rotational speed was 2586.3 rpm, the feed rate was 1055.2 mm/min, the cutting width was 1.6 mm and the cutting depth was 20 mm. Fig. 5.6 (b) illustrates the machining strategy. One workpiece was completed in four layers with one layer consisting of six cuts. Fig. 5.6 (c) shows an image of the workpiece after the machining of one layer. During machining, three tri-axial sensors were mounted inside the machine tool and used to collect force signals on the worktable and vibration signals on the worktable and spindle, respectively. The specifications of these sensors are given in Table 3.1. All signals were sampled at 51200 Hz.

The flank wear of the cutting tool was measured by a microscope after each cut as shown in Fig. 5.6 (d). All three cutting tools were used to complete the milling of six layers. The tool wear finally reached 395.3 μm, 421.1 μm and 490.0 μm, respectively. The images of flutes before and after machining are shown in Fig. 5.6 (e). According to [8], the allowable tool wear limit is set to 300 μm, after which the tool condition is considered faulty.

5.5.2 Monitoring results

The sensor data are segmented evenly based on the position of the tool. In total, around 2000 blocks of data matrices are generated for one cutting tool. After preprocessing using low-pass filtering and down-sampling, the data are used for process monitoring. The data from cuts in the first two layers are used as Phase I and the remaining four layers as Phase II. The linear interpolation is applied to the tool wear measurements such that the time when

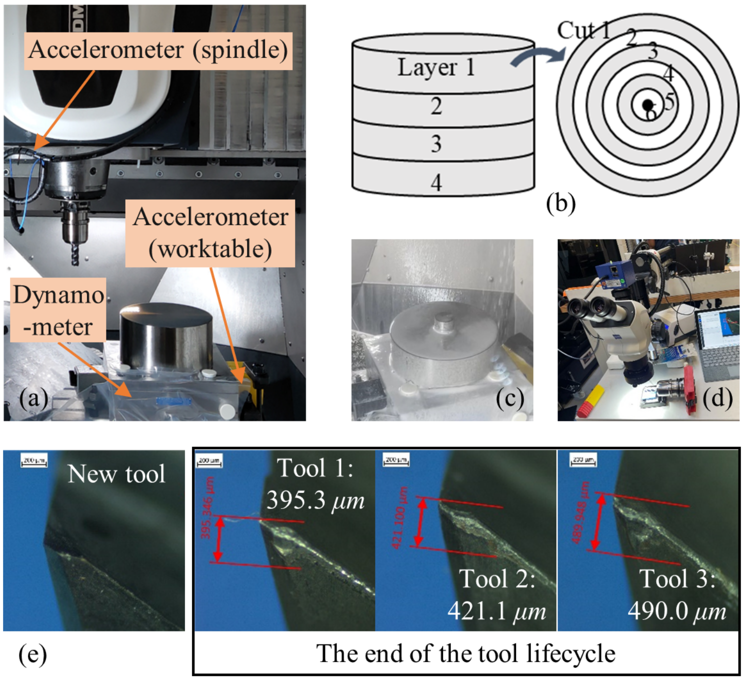


Figure 5.6: (a) Experimental setup and the sensors; (b) Diagram of the milling strategy; (c) The workpiece after the machining of one layer; (d) Measurement of tool wear; (e) The images of flutes before and after machining.

the flank wear reaches 300 μm can be located. For the proposed method, the forgetting factor η is 0.999 and the moving window length w is 16. The other methods are implemented with the same initialisation parameters as in the simulation study.

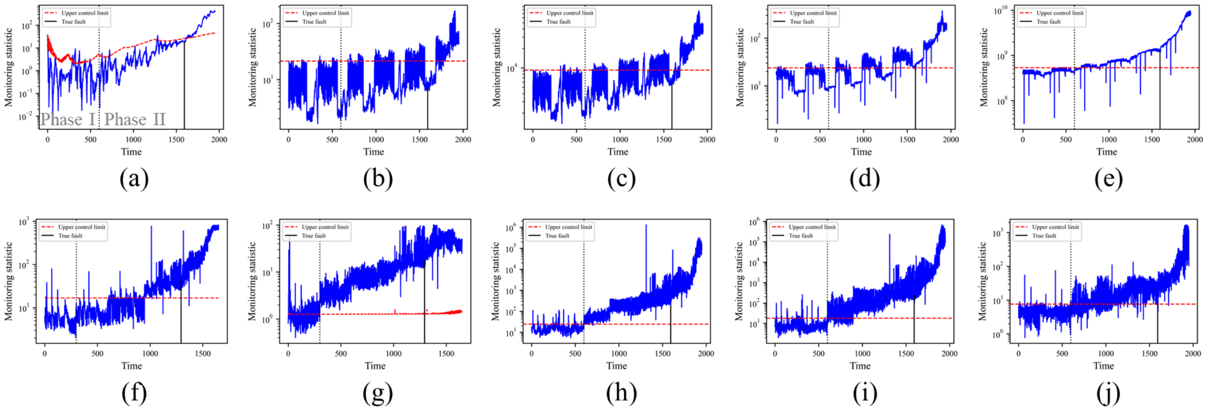


Figure 5.7: Monitoring results of Tool 1 using different methods: (a) RTSD T^2, (b) VPCA T^2, (c) VPCA SSE, (d) MPCA T^2, (e) MPCA SSE, (f) RPCA T^2, (g) RPCA Q, (h) SFA T_d^2, (i) SFA T_e^2, (j) CA T^2.

Figures 5.7-5.9 show the control charts for all three cutting tools based on different methods. The FDRs and FARs are listed in Table 5.2. It can be seen that almost all the statistics have an increasing trend due to the accumulation of tool wear. Compared with other statistics, T^2 in RTSD shows a curve with less fluctuation, with its adaptive control limit showing an increasing

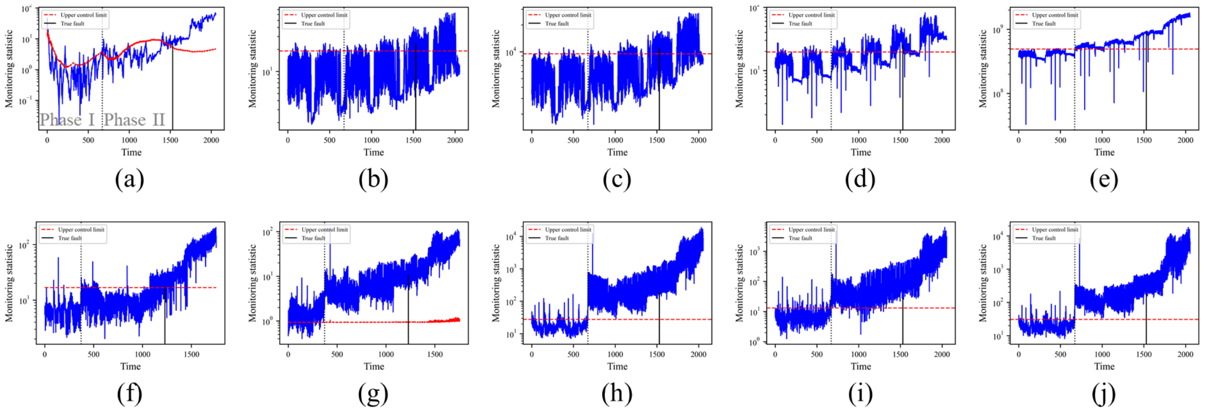


Figure 5.8: Monitoring results of Tool 2 using different methods: (a) RTSD T^2, (b) VPCA T^2, (c) VPCA SSE, (d) MPCA T^2, (e) MPCA SSE, (f) RPCA T^2, (g) RPCA Q, (h) SFA T_d^2, (i) SFA T_e^2, (j) CA T^2.

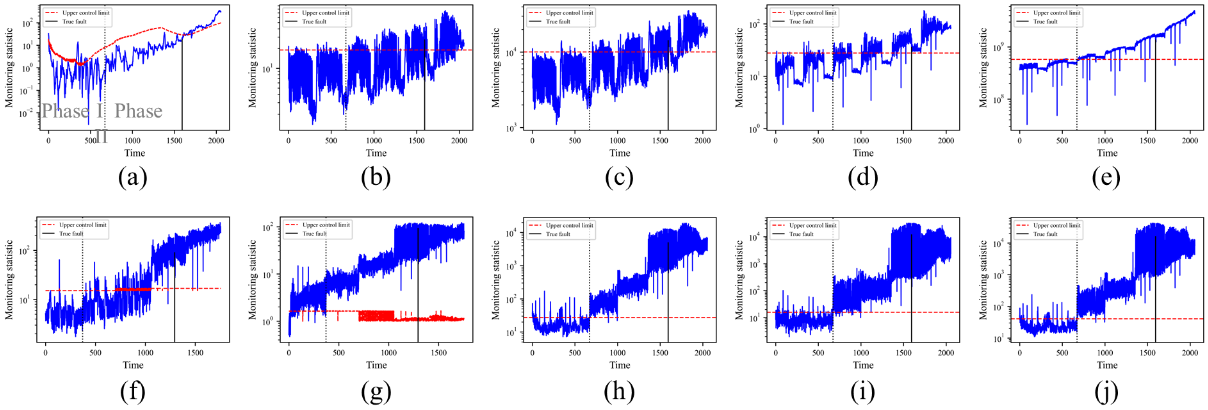


Figure 5.9: Monitoring results of Tool 3 using different methods: (a) RTSD T^2, (b) VPCA T^2, (c) VPCA SSE, (d) MPCA T^2, (e) MPCA SSE, (f) RPCA T^2, (g) RPCA Q, (h) SFA T_d^2, (i) SFA T_e^2, (j) CA T^2.

trend. In Phase II, the statistics are below the limits initially and exceed the limits when the tool condition becomes severely worn. This result demonstrates that the proposed adaptive control limit can achieve a good trade-off between FDR and FAR. As shown in Table 5.2, the RTSD method can achieve very high FDR and low FAR, especially for tool 1 and 3. For tool 2, even though the FAR with the proposed method is not the lowest, the FDR with the new method is 100% compared to 94% with the RPCA method, which has the lowest FAR.

In the control charts of VPCA and MPCA, the change in statistics displays six repeated patterns corresponding to six layers. After the happening of tool fault, the statistic reaches a higher level at a higher rate, especially for tool 1. However, the UCL determined in Phase I cannot distinguish these two states successfully. The VPCA method tends to produce low FDR and low FAR, while the MPCA method shows both high FDR and high FAR. In the control charts of the other three methods, the utilisation of signal features weakens the existence of repeated patterns. But the statistic with SFA and CA shows a step change at the initial

Table 5.2: Monitoring results in the experimental study

Tool	Metrics	RTSD T^2	VPCA T^2	VPCA SSE	MPCA T^2	MPCA SSE	RPCA T^2	RPCA Q	SFA T_d^2	SFA T_e^2	CA T^2
1	FDR(%)	91.6	68.2	78.2	98.6	99.7	99.7	**100.0**	**100.0**	**100.0**	99.4
	FAR(%)	**2.0**	16.9	27.4	46.5	88.9	43.0	100.0	99.4	90.8	69.8
2	FDR(%)	**100.0**	42.4	49.6	78.1	99.6	94.1	**100.0**	**100.0**	**100.0**	**100.0**
	FAR(%)	25.8	18.9	18.5	34.5	92.7	**15.5**	100.0	100.0	95.0	100.0
3	FDR(%)	87.9	59.5	76.8	99.8	99.8	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
	FAR(%)	**0.0**	33.4	37.1	47.4	96.6	40.0	100.0	99.8	98.0	99.8

stage of Phase II, which results in all the samples being identified as faulty. Even though the adaptive control limit is used in RPCA, it still cannot adapt to the normal increase of tool wear, leading to a very high FAR. Especially, the Q statistic fails to detect any fault. Only the T^2 statistic performs well in Tool 2, achieving a similar overall accuracy as the proposed method. Therefore, the overall monitoring performance of other methods that have been applied for comparison is not reliable.

5.5.3 Discussion

To demonstrate the effectiveness of the recursive decomposition strategy, the ordinary TSD algorithm introduced in previous chapters is applied to analyse the experimental data. Figures 5.10 (a)-(f) show the heatmaps of SDC1s obtained using RTSD and TSD algorithms for all three tools. The TSD-yielded SDC1s exhibit repeated patterns corresponding to repeated cutting passes. In the later milling period, the SDCs show more fluctuations due to excessive tool wear. However, the RTSD-yielded SDC1s change smoothly, capturing general trend of ordinary SDC1s with unexpected fluctuations eliminated. Figures 5.10 (g)-(i) demonstrate the control charts based on TSD method. The calculation of monitoring statistics and adaptive UCL follows the same procedure mentioned above, apart from TDC1s and SDC1s are generated through TSD algorithm. In comparison with Figures 5.7 (a)-5.9 (a), the TDC1 features-based T^2 statistic has a similar trend, but the SDC1s-based UCL in TSD method fluctuates greatly and do not reflect process trend. The results validate the importance of the proposed recursive TSD method.

The influence of key parameters including forgetting factor η and window length w on the monitoring results has been studied. Table 5.3 presents the monitoring results with different values of η and w. When η is selected as 0, the decomposition method becomes ordinary TSD. Increasing η and w can smoothen the curves of control limit and monitoring statistics, re-

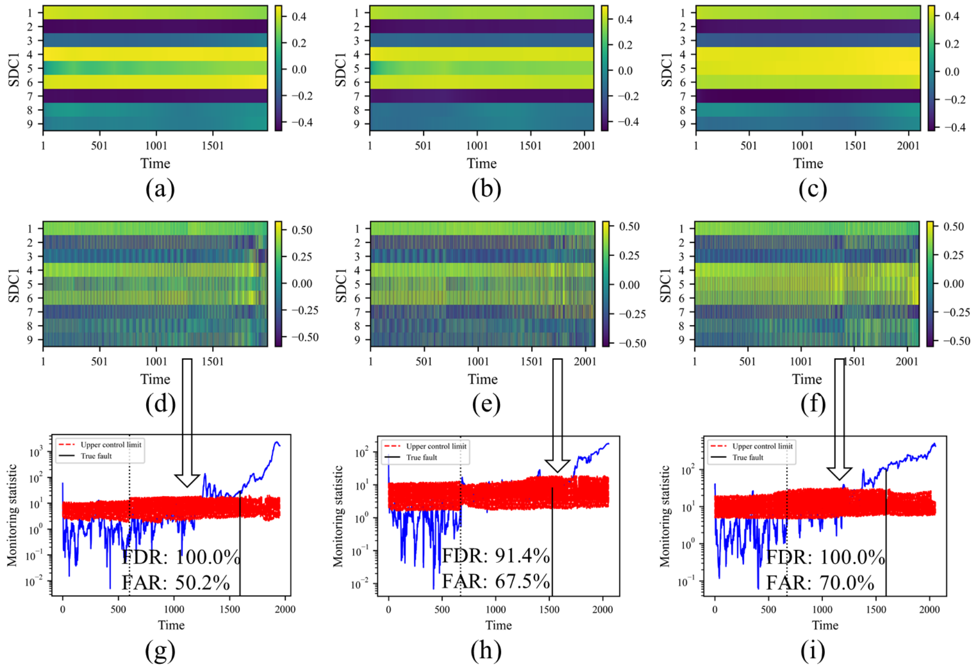


Figure 5.10: Comparison of SDC1s obtained using RTSD and TSD algorithms in the experimental study: (a) RTSD Tool 1, (b) RTSD Tool 2, (c) RTSD Tool 3, (d) TSD Tool 1, (e) TSD Tool 2, (f) TSD Tool 3; Monitoring results using TSD T^2: (g) Tool 1, (h) Tool 2, (i) Tool 3.

spectively. Considering the principle of the proposed method, capturing slow and rapid trend of underlying processes via sensor and time domain components, the value of η should be large enough while w should avoid too big values. Otherwise the SDCs fluctuate significantly or the time domain features reflect slow trend of process changes as well, leading to high false alarm rate as shown in Table 5.3. The sensitivity study further explains the essential of the proposed method. Future research will focus on the selection of these parameters to realise an optimal trade-off between FDR and FAR.

5.6 Conclusion

In this chapter, a novel time-varying process monitoring method based on a recursive time-sensor decomposition framework has been proposed. Instead of applying TSD to the data blocks separately as shown in previous chapters, RTSD considers the time-dependency among all available data and updates the correlation matrix recursively. Therefore, the obtained SDCs can reflect the unknown but deterministic trend of the process. Meanwhile, the feature vectors are extracted from TDCs. The moving average of these feature vectors is used as the monitoring sample to calculate the T^2 statistic to deal with possible fluctuation problems. Then, the distribution of SDCs is utilised to determine an adaptive control limit, which can

Table 5.3: The influence of η and w on the monitoring results in the experimental study

Tool	Metrics	Forgetting factor η ($w = 16$)				Window length w ($\eta = 0.999$)			
		0 (TSD)	0.9	0.99	**0.999**	8	**16**	64	128
1	FDR(%)	100.0	99.4	45.3	91.6	74.0	91.6	89.1	100.0
	FAR(%)	50.2	45.2	2.6	2.0	1.8	2.0	0.0	26.8
2	FDR(%)	91.4	99.6	100.0	100.0	100.0	100.0	100.0	100.0
	FAR(%)	67.5	28.3	22.1	25.8	20.7	25.8	37.6	32.6
3	FDR(%)	100.0	95.2	15.8	87.9	50.9	87.9	76.8	75.8
	FAR(%)	70.0	26.4	1.4	0.0	0.1	0.0	0.0	0.0

adapt to normal time-varying behaviour of the process. The proposed approach has been validated through a simulation study and an experimental study on the tool wear process during dynamic milling. Comparative studies with existing methods have also been conducted. The results have demonstrated that the new approach can provide the best trade-off between FDR and FAR. Early false alarms can be avoided, providing a reliable monitoring performance.

6

Feature selection based on local regularisation assisted SALSA and application to tool wear prediction

6.1 Introduction

This chapter focuses on another important topic in process monitoring, which is the prediction of process states such as the volume of production and the RUL of key components. Accurate prediction forms the basis of a prognostic and health management system. In advanced manufacturing, tool wear prediction involves forecasting the gradual degradation of cutting tools over time. It can be considered more ambitious and challenging to realise compared to breakage detection and wear monitoring that are realised in previous chapters. Because tool wear can be influenced by many factors and often exhibit nonlinear behaviour. Besides, real-time implementation requires fast and efficient computational algorithms capable of processing large volumes of sensor data. To ensure the completeness of the research about PHM of cutting tools, the problem of tool wear prediction is studied in this chapter.

Data-driven process prediction techniques have been widely applied in complex industrial systems [32, 195]. A regression model is typically used to predict process states from the multi-sensor measurements [196, 197]. As mentioned in Chapter 2, a large number of features can be extracted and used as inputs of the regression model. When the dimension of available data or features becomes extremely high, a critical issue known as the curse of dimensionality, arises in the data-mining step, leading to overfitting and huge computational costs [198]. Besides, it is common that multiple sensors measure similar aspects of the process, bringing about the problem of multicollinearity. The estimation of model coefficients could be very unstable, making the model less robust to noise in the data. To build an accurate condition monitoring model, it is essential to develop an effective and efficient feature selection method to select a subset of important features and address the challenges mentioned above.

Feature selection has shown necessity and benefits in a wide range of areas including machine learning, pattern recognition, multimedia, and condition monitoring [199, 200]. Its main objective is to select the most important features from all candidates such that a simpler and more comprehensible model is built and the data-mining performance is improved [201]. As mentioned in Chapter 3, feature selection methods can be broadly categorised as filter, wrapper, and embedded methods. Among them, embedded methods provide a trade-off between filter and wrapper methods, which integrate the feature selection into model learning. The introduction of learning algorithms improves the performance of selected features. Besides, compared with wrapper methods, they are far more efficient as the repeated feature evaluation procedure is no longer needed. Hence, the embedded method will be the main focus of this chapter.

The regularisation technique is a representative embedded feature selection method that aims to build a sparse model by minimising the fitting errors and forcing model coefficients to be small enough at the same time [92]. To achieve this, the l_p-norm penalty term is added on a machine learning model, where $0 \leq p \leq 1$. When $p = 0$, the l_0-norm penalises the number of non-zero entries of the coefficient vector. Such an integer programming problem is usually difficult to solve. Therefore, the l_1-norm regularisation is usually used as a relaxation of the l_0-norm. In this case, many feature coefficients become smaller but not exactly zero. The l_1-norm regularised feature selection method has attracted lots of attention in recent years [202]. Furthermore, to cope with feature selection requirements in multi-class classification or multivariate regression problems, the regularisation method is improved by adding the $l_{p,q}$-norm penalty term, where $p > 1$ and $0 \leq p \leq 1$. Different settings of p and q provide fine adjustments of the regularisation effect [203, 204]. Researchers have also proposed some feature selection methods for specific models, for example, the orthogonal forward selection (OFR) method is usually exploited in linear regression problems [205] and the automatic relevance determination kernel functions are often used in feature selection of Bayesian regression problems [95].

As mentioned above, the l_1-norm optimisation method is an attractive topic in feature selection and sparse model estimation. These methods aim to minimise not only the prediction error but also the sum of the absolute values of the model coefficients [206]. One of the most popular l_1-norm optimisation algorithms is the least absolute shrinkage and selection operator (LASSO), which can be used to prune redundant features for condition monitoring. In [207], the authors showed that LASSO had the smallest tool wear prediction error compared with methods such as FDR, ridge regression, and principal component regression. However, extensive research shows that LASSO often produces an insufficiently sparse model, leading to inaccurate prediction results [206]. To achieve a more sparse and accurate solution, the split augmented Lagrangian shrinkage algorithm (SALSA) is proposed. SALSA converts the l_1-norm optimisation problem into several subproblems, which can be addressed separately without the need for a third-party solver [208]. Based on SALSA, recently, a new approach called sparse augmented Lagrangian (SAL) has been developed to build more compact models [206]. SAL combines the random subsampling technique with the SALSA method to produce intermediate models. Then, the highly selected terms from these intermediate models are added to the final model. Finally, SALSA is applied again to estimate the model coefficients. This leads to a better performance in model sparsity compared with LASSO and SALSA. In [117], the SAL method was used to select the significant features for tool wear prediction in machining. However, SAL suffers from low computational efficiency due to the application of the random subsampling technique, which involves conducting numerous repetitions of the SALSA algorithm. In [209], the authors proposed a re-weighted l_1-minimisation algorithm under the sparse Bayesian learning (SBL) framework. Nevertheless, the solution has to be calculated using a third-party solver. To cope with this limitation, the Bayesian augmented Lagrangian (BAL) algorithm is developed [210], which solves the l_1-minimisation problem in SBL using the SALSA method. Even though the computation efficiency has been improved, BAL still takes much longer time than LASSO and SALSA algorithms. From the analysis mentioned above, it is still an open question of developing a l_1-norm optimisation-based feature selection method with high sparsity, fast computation speed, and high prediction accuracy.

To fundamentally address the challenges, this chapter proposes anovel algorithm referred to as local regularisation assisted SALSA (LR-SALSA). The key idea is to embed a local regularisation strategy into the framework of SALSA. Firstly, the l_1-minimisation problem is converted to three subproblems that can be solved separately. The first subproblem is a ridge regression problem, which does not involve the pruning of redundant terms originally. In LR-SALSA, the local regularisation strategy is employed to solve the first subproblem, which means that each candidate term has an associated individual regularisation parameter. During the optimisation process, many of these penalty parameters are driven to large values, resulting in a successive removal of corresponding terms from the pool of candidate terms. This strategy improves the computational efficiency and enhances the sparsity of the final model by applying SALSA only once. After determining the important terms, the model coefficients are re-estimated using the least squares method. As a result, the main contributions of this study are as follows,

1. The LR-SALSA method is proposed under the Bayesian evidence framework, introducing an individual penalty parameter for each coefficient of the model. An iterative algorithm is derived to optimise these hyperparameters relying on the dataset only based on the Bayesian evidence framework [211]. In practice, this optimisation process shows a quicker convergence than method based on a third-party solver like BAL.

2. The optimisation of local regularisation hyperparameters is embedded into the iterative algorithm of SALSA. At each step, according to the values of the penalty parameters, redundant variables can be pruned in time, which significantly reduces the overall computing complexity. As a result, LR-SALSA exhibits better model sparsity and higher computational efficiency when compared to existing l_1-norm optimisation algorithms, such as SAL and BAL.

To validate the performance of LR-SALSA, both simulation and experimental studies have been conducted. The results of these studies show that LR-SALSA successfully produces a more compact model while maintaining a high level of prediction accuracy and fast computation speed. This demonstrates the effectiveness and practicality of the proposed feature selection method in real-world condition monitoring applications.

6.2 Theoretical background

6.2.1 Problem description

Consider the following l_1-norm optimisation problem

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^M} \quad f_1(\boldsymbol{\theta}) + f_2(\boldsymbol{\theta}) \tag{6.1}$$

with $f_1(\boldsymbol{\theta}) = \frac{1}{2}\|\boldsymbol{y} - \mathbf{P}\boldsymbol{\theta}\|_2^2$ and $f_2(\boldsymbol{\theta}) = \lambda\|\boldsymbol{\theta}\|_1$, where $\boldsymbol{y} \in \mathbb{R}^N, \mathbf{P} \in \mathbb{R}^{N\times M}$ represent the output and regression matrix composed of multivariate inputs, respectively. λ denotes the l_1 regularisation parameter. To convert the original problem to a constrained problem, the coefficient vector $\boldsymbol{\theta}$ in function f_2 is replaced with a new variable $\boldsymbol{\nu}$, which gives

$$\begin{aligned} \min_{\boldsymbol{\theta},\boldsymbol{\nu}\in\mathbb{R}^M} \quad & f_1(\boldsymbol{\theta}) + f_2(\boldsymbol{\nu}) \\ \text{s.t.} \quad & \boldsymbol{\theta} = \boldsymbol{\nu} \end{aligned} \tag{6.2}$$

In order to find the optimal solution to this problem, the Lagrangian function can be defined as

$$L(\boldsymbol{\theta}, \boldsymbol{\nu}, \boldsymbol{\phi}) = f_1(\boldsymbol{\theta}) + f_2(\boldsymbol{\nu}) + \boldsymbol{\phi}^T(\boldsymbol{\theta} - \boldsymbol{\nu}) \tag{6.3}$$

where $\boldsymbol{\phi} \in \mathbb{R}^M$ is the Lagrangian multiplier. The solution to Eq. (6.2) is always a saddle point of $L(\boldsymbol{\theta}, \boldsymbol{\nu}, \boldsymbol{\phi})$ in (6.3).

Typically, the dual ascent algorithm can be used to optimise the primal and dual problems, respectively. However, this algorithm may not always converge, especially when applied to non-convex or non-smooth optimisation problems. So the Augmented Lagrangian method (ALM) is introduced by incorporating a penalty term

$$L_\mu(\boldsymbol{\theta}, \boldsymbol{\nu}, \boldsymbol{\phi}) = f_1(\boldsymbol{\theta}) + f_2(\boldsymbol{\nu}) + \boldsymbol{\phi}^T(\boldsymbol{\theta} - \boldsymbol{\nu}) + \frac{\mu}{2}\|\boldsymbol{\theta} - \boldsymbol{\nu}\|_2^2 \tag{6.4}$$

where $\mu \geq 0$ denotes the penalty parameter. The penalty term helps to enforce the constraints, leading to a more stable and efficient algorithm. Compared with the dual ascent method, the convergence is guaranteed without assumptions like strict convexity of objective functions.

Then, considering sometimes the objective function is separable, the alternating direction method of multipliers (ADMM) algorithm [212] is proposed. The primal variable optimisation step can be further split into subproblems that can be solved in parallel. Let $\boldsymbol{d} = \boldsymbol{\phi}/\mu$, with ADMM algorithm, problem (6.4) is solved by optimising the following subproblems,

$$\hat{\boldsymbol{\theta}}_{k+1} = \arg\min_{\boldsymbol{\theta}} f_1(\boldsymbol{\theta}) + \frac{\mu}{2}\|\boldsymbol{\theta} - \boldsymbol{\nu}_k + \boldsymbol{d}_k\|_2^2 \tag{6.5}$$

$$\boldsymbol{\nu}_{k+1} = \arg\min_{\boldsymbol{\nu}} f_2(\boldsymbol{\nu}) + \frac{\mu}{2}\|\hat{\boldsymbol{\theta}}_{k+1} - \boldsymbol{\nu} + \boldsymbol{d}_k\|_2^2 \tag{6.6}$$

$$\boldsymbol{d}_{k+1} = \boldsymbol{d}_k + (\hat{\boldsymbol{\theta}}_{k+1} - \boldsymbol{\nu}_{k+1}) \tag{6.7}$$

6.2.2 SALSA method

The SALSA algorithm can be regarded as an application of the ADMM algorithm to the l_1-norm regularised linear regression problem (6.1). It converts the original problem into several subproblems as shown in (6.5)-(6.7). Then these problems are addressed separately as shown in **Algorithm 1**. To elaborate further,

- For (6.5), $\hat{\boldsymbol{\theta}}_{k+1}$ can be calculated by taking the derivative of the cost function with respect to $\boldsymbol{\theta}$ to be zero.

- Concerning (6.6), the update for $\boldsymbol{\nu}_{k+1}$ incorporates the utilisation of the soft thresholding operator $S_{\lambda/\mu}(z) = \max(0, z - \lambda/\mu) - \max(0, -z - \lambda/\mu)$ [213], where $z = \hat{\boldsymbol{\theta}}_{k+1} + \boldsymbol{d}_k$.

- At each iteration, the sum-of-squares error is evaluated through $e_{k+1} = (\boldsymbol{y} - \mathbf{P}\hat{\boldsymbol{\theta}}_{k+1})^T(\boldsymbol{y} - \mathbf{P}\hat{\boldsymbol{\theta}}_{k+1})$. The iteration stops when the sequence of errors converges.

Algorithm 1 SALSA

```
Predefine $\mu$, $\lambda$, and $\delta_e$; set $\boldsymbol{\nu}_0 = \boldsymbol{d}_0 = \mathbf{0}$
Repeat
    $\hat{\boldsymbol{\theta}}_{k+1} = (\mathbf{P}^T\mathbf{P} + \mu\mathbf{I})^{-1}(\mathbf{P}^T\boldsymbol{y} + \mu(\boldsymbol{\nu}_k - \boldsymbol{d}_k))$
    $\boldsymbol{\nu}_{\boldsymbol{k+1}} = \max(0, \hat{\boldsymbol{\theta}}_{k+1} + \boldsymbol{d}_k - \lambda/\mu) - \max(0, -(\hat{\boldsymbol{\theta}}_{k+1} + \boldsymbol{d}_k) - \lambda/\mu)$
    $\boldsymbol{d}_{k+1} = \boldsymbol{d}_k + (\hat{\boldsymbol{\theta}}_{k+1} - \boldsymbol{\nu}_{k+1})$
    $e_{k+1} = (\boldsymbol{y} - \mathbf{P}\hat{\boldsymbol{\theta}}_{k+1})^T(\boldsymbol{y} - \mathbf{P}\hat{\boldsymbol{\theta}}_{k+1})$
    $k \leftarrow k + 1$
Until stopping criterion $|e_{k+1} - e_k| \leq \delta_e$ is satisfied
```

6.2.3 SAL method

The SAL method, inspired by SALSA, aims to construct a parsimonious model. This approach encompasses two main stages: variable selection and parameter estimation.

- During the variable selection stage, the random subsampling technique is used to randomly generate multiple data subsets, each comprising system output $\widetilde{\boldsymbol{y}}_j$ and input $\widetilde{\mathbf{P}}_j$.

For each subsampled dataset, SALSA executes the procedure outlined in **Algorithm 1**, resulting in an intermediate model. The terms that are present frequently in all intermediate models will be selected as the dominant terms.

- Subsequently, in the parameter estimation stage, SALSA will be executed again on a subsampled dataset with the selected important terms. This step determines the coefficients of the final model.

Within this algorithm, both pruning and stability selection play vital roles in ensuring model sparsity. While the pruning process removes terms with minimal impact on model performance based on the subsampling dataset, stability selection discards redundant terms. The comprehensive procedure of the SAL method is presented in **Algorithm 2**.

6.3 Methodology

6.3.1 Feature dictionary construction

Assume the whole process to be monitored is divided into N stages and the number of available variables is I. The multi-sensor data from the nth stage is denoted as $\mathbf{X}^n = [\boldsymbol{x}_1^n, \boldsymbol{x}_2^n, ..., \boldsymbol{x}_I^n]$, where $\boldsymbol{x}_i^n$ is the ith variable. From $\boldsymbol{x}_i^n$, J representative features can be extracted and denoted as a row vector $\boldsymbol{p}_i^n = [p_{i,1}^n, p_{i,2}^n, ..., p_{i,J}^n]$. Concatenating the features vectors of all I variables yields an vector $\boldsymbol{p}^n = [\boldsymbol{p}_1^n, \boldsymbol{p}_2^n, ..., \boldsymbol{p}_I^n] \in \mathbb{R}^{1\times IJ}$.

To enhance the information content of the extracted features, the feature enlarging procedure is applied. For example, polynomial combinations of the features can add nonlinear terms to the dictionary matrix and represent the nonlinear relationship between the target and variables. Here, for the simplicity of notations, $\boldsymbol{p}^{\mathrm{I}} = [p_1, p_2, ..., p_{IJ}]$ is used to denote the original feature vector of any data block. Obtaining terms in the kth order polynomial set is identical to finding out combinations with repetition. The number of possible selections is given by $\binom{IJ+k-1}{k} = \frac{(IJ+k-1)!}{k!(IJ-1)!}$. For example, the second-order feature vector is as follows,

$$\boldsymbol{p}^{\mathrm{II}} = [p_1^2, p_1 \cdot p_2, ..., p_1 \cdot p_{IJ}, p_2^2, ..., p_{IJ-1} \cdot p_{IJ}, p_{IJ}^2] \tag{6.8}$$

where the dimension of $\boldsymbol{p}^{\mathrm{II}}$ is $1 \times \frac{IJ(IJ+1)}{2}$.

Suppose the highest degree is K, the extended feature vector is given by

$$\boldsymbol{p} = [\boldsymbol{p}^{\mathrm{I}}, \boldsymbol{p}^{\mathrm{II}}, ..., \boldsymbol{p}^{\mathrm{K}}] \in \mathbb{R}^{1\times M} \tag{6.9}$$

Algorithm 2 SAL

```
Variable selection stage:
1.  Predefine $\mu$, $\lambda$, $\delta_\theta$, and $\delta_{thr}$; set $\boldsymbol{v}_0 = \boldsymbol{d}_0 = \boldsymbol{0}$
2.  for $j = 1$ to $n$:
3.      Random subsampling $\longrightarrow \widetilde{\boldsymbol{y}}_j$ and $\widetilde{\mathbf{P}}_j$
4.      Run SALSA and stop when $sign(\hat{\boldsymbol{\theta}}_{k+1}) = sign(\hat{\boldsymbol{\theta}}_k)$
5.      if $\hat{\theta}_i^2/||\hat{\boldsymbol{\theta}}||_2^2 \leq \delta_\theta$ then
6.          Term $\boldsymbol{p}_i$ is removed
7.      end if
8.      Record the selected terms of the intermediate model
9.  end for
10. if $sp(\boldsymbol{p}_i)/n \geq \delta_{thr}$ where $sp(\boldsymbol{p}_i)$ is the occurrence of $\boldsymbol{p}_i$ then
11.     Term $\boldsymbol{p}_i$ is selected
12. end if
13. Choose the mostly selected terms as the final model
Parameter estimation stage:
12. Random subsampling $\longrightarrow \widetilde{\boldsymbol{y}}$ and $\widetilde{\mathbf{P}}_s$
13. Run SALSA and stop when $sign(\hat{\boldsymbol{\theta}}_{k+1}) = sign(\hat{\boldsymbol{\theta}}_k)$
14. Determine the coefficients in the final model
```

where $M = IJ + \frac{IJ(IJ+1)}{2} + \ldots + \frac{(IJ+K-1)!}{K!(IJ-1)!}$. Concatenating the extended feature vectors of all N stages produces the feature dictionary

$$\mathbf{P} = \begin{bmatrix} \boldsymbol{p}^{1,\mathrm{I}} & \boldsymbol{p}^{1,\mathrm{II}} & \cdots & \boldsymbol{p}^{1,\mathrm{K}} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{p}^{N,\mathrm{I}} & \boldsymbol{p}^{N,\mathrm{I}} & \cdots & \boldsymbol{p}^{N,\mathrm{K}} \end{bmatrix} \in \mathbb{R}^{N \times M} \tag{6.10}$$

Note that $\mathbf{P}$ is an overcomplete dictionary with $M \gg N$. Let $\boldsymbol{y} \in \mathbb{R}^{N \times 1}$ denote the vector of prediction targets. The regression problem is a typical sparse coding problem, which finds an optimal sparse vector $\boldsymbol{\theta}$ such that $\boldsymbol{y} - \mathbf{P}\boldsymbol{\theta}$ is minimised. In this study, the constraint of the solution vector takes the form of l_1 norm, the process prediction problem is exactly the l_1 minimisation problem (6.1).

6.3.2 Solving the subproblems

The original l_1-norm optimisation problem (6.1), is converted to subproblems (6.5)-(6.7) using the ADMM framework. In solving subproblem (6.5), a local regularisation strategy is employed to penalise each element of $\boldsymbol{\theta}$ for the purpose of producing a more compact model. Let $\boldsymbol{e} = \boldsymbol{y} - \mathbf{P}\boldsymbol{\theta}$, $\boldsymbol{\eta} = \boldsymbol{\nu} - \boldsymbol{d}$, the new cost function is written as

$$J_R(\boldsymbol{\theta}) = \boldsymbol{e}^T\boldsymbol{e} + \sum_{i=1}^{M} \mu_i \left(\theta_i - \eta_i\right)^2 \tag{6.11}$$

where $\mu_i \geq 0$ is the penalty parameter of element θ_i in $\boldsymbol{\theta}$. Compared with the 'global' regularisation method, where a uniform intensity of penalty is applied to all elements in $\boldsymbol{\theta}$ in SALSA and SAL, this novel design of 'local' regularisation assisted SALSA allows independent and flexible control of the penalty intensity of every coefficient in the optimisation problem. Thus the function (6.11) consists of the sum-of-squares error function and a so-called locally quadratic regularisation term.

The Bayesian evidence approximation can be used to optimise the regularisation parameters μ_i automatically. From the Bayesian viewpoint, a regularisation parameter is equivalent to the ratio of the coefficient precision and the noise precision [211]. In this sense, $J_R(\boldsymbol{\theta})$ is equivalent to the following cost function,

$$J_B(\boldsymbol{\theta}) = \beta\boldsymbol{e}^T\boldsymbol{e} + \sum_{i=1}^{M} \alpha_i \left(\theta_i - \eta_i\right)^2 = \beta\boldsymbol{e}^T\boldsymbol{e} + (\boldsymbol{\theta} - \boldsymbol{\eta})^T\mathbf{A}(\boldsymbol{\theta} - \boldsymbol{\eta}) \tag{6.12}$$

where β and α_i govern the precision parameters of noise and θ_i, and $\mathbf{A} = \mathrm{diag}\{\alpha_1, \alpha_2, ..., \alpha_M\}$ denotes the precision matrix. The regularisation parameter in (6.11) has a relationship with the hyperparameters as $\mu_i = \alpha_i/\beta$.

Using the evidence approximation framework [214], the unknown parameters $\boldsymbol{\alpha}$ and β are determined by maximising the log marginal likelihood function

$$\begin{aligned} &\ln p(\boldsymbol{y}|\mathbf{P}, \boldsymbol{\eta}, \boldsymbol{\alpha}, \beta) \\ =&\frac{N}{2}\ln\frac{\beta}{2\pi} + \frac{1}{2}\sum_{i=1}^{M}\ln\alpha_i - \frac{1}{2}\beta\boldsymbol{y}^T\boldsymbol{y} - \frac{1}{2}\boldsymbol{\eta}^T\mathbf{A}\boldsymbol{\eta} + \frac{1}{2}\boldsymbol{m}^T\mathbf{S}^{-1}\boldsymbol{m} + \frac{1}{2}\ln|\mathbf{S}| \end{aligned} \tag{6.13}$$

where $\boldsymbol{m}$ and $\mathbf{S}$ define the mean and covariance matrix of the posterior distribution for $\boldsymbol{\theta}$, and are given by

$$\boldsymbol{m} = \mathbf{S}(\mathbf{A}\eta + \beta\mathbf{P}^T\boldsymbol{y}) \tag{6.14}$$

$$\mathbf{S} = (\mathbf{A} + \beta\mathbf{P}^T\mathbf{P})^{-1} \tag{6.15}$$

Taking the derivatives of $\ln p(\boldsymbol{y}|\mathbf{P}, \boldsymbol{\eta}, \boldsymbol{\alpha}, \beta)$ with respect to $\boldsymbol{\alpha}$ and β to be zero yields the updating formulas for $\boldsymbol{\alpha}$ and β, respectively.

$$\alpha_i^{\text{new}} = \frac{\gamma_i}{(m_i - \eta_i)^2} \tag{6.16}$$

$$(\beta^{\text{new}})^{-1} = \frac{\|\boldsymbol{y} - \mathbf{P}\boldsymbol{m}\|_2^2}{N - \sum_{i=1}^{M}\gamma_i} \tag{6.17}$$

where $\gamma_i = 1 - \alpha_i S_{ii}$, S_{ii} is the ith diagonal component of $\mathbf{S}$. The detailed derivation of (6.16) and (6.17) is presented in the Appendix.

The optimisation begins by setting initial values for $\boldsymbol{\alpha}$ and β. Using these initial values, the posterior mean and covariance are computed according to (6.14) and (6.15), respectively. Subsequently, the hyperparameters are re-estimated using (6.16) and (6.17); and re-estimating the posterior mean and covariance, using (6.14) and (6.15), until a suitable convergence criterion is satisfied. It is worth mentioning that the optimisation results in a proportion of α_i that are close to large values, and so the coefficients θ_i corresponding to these hyperparameters have posterior distributions with a mean of η_i and variance of zero. Thus those variables are removed from the model and a sparse structure is found.

Giving values $\boldsymbol{\alpha}_k^*$ and β_k^* for the hyperparameters that maximise the marginal likelihood and solutions of $\boldsymbol{\nu}_k$ and $\boldsymbol{d}_k$, at the $(k+1)$th iteration, $\boldsymbol{\alpha}_k^*$ and β_k^* are set as the initial values, and $\boldsymbol{\eta}_k = \boldsymbol{\nu}_k - \boldsymbol{d}_k$. Running the optimisation procedure yields optimal values $\boldsymbol{\alpha}_{k+1}^*$ and β_{k+1}^*. At the same time, $\hat{\boldsymbol{\theta}}_{k+1}$ and $\boldsymbol{\mu}_{k+1}$ are updated

$$\hat{\boldsymbol{\theta}}_{k+1} = (\mathbf{A}_{k+1}^* + \beta_{k+1}^*\mathbf{P}^T\mathbf{P})^{-1}(\mathbf{A}_{k+1}^*\eta + \beta_{k+1}^*\mathbf{P}^T\boldsymbol{y}) \tag{6.18}$$

$$\mu_{i,k+1} = \alpha_{i,k+1}/\beta_{k+1} \tag{6.19}$$

Similar to SALSA, the subproblem (6.6) is also solved using the soft thresholding operator $S_{\lambda/\mu}(z)$. Each element of the variable $\boldsymbol{\nu}_{k+1}$ is updated as

$$\nu_{i,k+1} = \max(0, \hat{\theta}_{i,k+1} + d_{i,k} - \lambda/\mu_{i,k+1}) - \max(0, -\hat{\theta}_{i,k+1} - d_{i,k} - \lambda/\mu_{i,k+1}) \tag{6.20}$$

With $\hat{\boldsymbol{\theta}}_{k+1}$ and $\boldsymbol{\nu}_{k+1}$, $\boldsymbol{d}_{k+1}$ can be calculated by

$$\boldsymbol{d}_{k+1} = \boldsymbol{d}_k + (\hat{\boldsymbol{\theta}}_{k+1} - \boldsymbol{\nu}_{k+1}) \tag{6.21}$$

In terms of the stopping criterion, the sum-of-squares error is computed by $e_{k+1} = (\boldsymbol{y} - \mathbf{P}\hat{\boldsymbol{\theta}}_{k+1})^T(\boldsymbol{y} - \mathbf{P}\hat{\boldsymbol{\theta}}_{k+1})$ and the iteration stops when $|e_{k+1} - e_k| \le \delta_e$, where δ_e denotes the convergence tolerance.

6.3.3 Variable selection

The pruning of variables is carried out at each iterative step. Initially, $\boldsymbol{\eta}$ equals zero, so the terms that have very small estimated coefficients, i.e., $\hat{\theta}_i \longrightarrow 0$, will be removed. This is because corresponding hyperparameters α_i tend to become extremely large. At the $(k+1)$th iteration, if the estimated coefficients are close to $\boldsymbol{\eta}_k$, i.e., $\hat{\theta}_{i,k+1} \longrightarrow \eta_{i,k}$, the corresponding terms will be removed. This operation can guarantee the convergence of optimising hyperparameters $\boldsymbol{\alpha}_{k+1}$ and β_{k+1}. Besides, pruning helps to build a sparse model. Intuitively, the terms with coefficients that converge to $\boldsymbol{\eta}_k$ will not be selected. Substituting $\boldsymbol{\nu}_k$ in the expression of $\boldsymbol{\eta}_k$ with (6.20), the following equation holds

$$\eta_{i,k} = \begin{cases} \hat{\theta}_{i,k} + d_{i,k-1} - 2\lambda/\mu_{i,k}, & \hat{\theta}_{i,k} + d_{i,k-1} > \lambda/\mu_{i,k} \\ -\hat{\theta}_{i,k} - d_{i,k-1}, & |\hat{\theta}_{i,k} + d_{i,k-1}| \le \lambda/\mu_{i,k} \\ \hat{\theta}_{i,k} + d_{i,k-1} + 2\lambda/\mu_{i,k}, & \hat{\theta}_{i,k} + d_{i,k-1} < -\lambda/\mu_{i,k} \end{cases} \tag{6.22}$$

In principle, if $|\hat{\theta}_{i,k} + d_{i,k-1}|$ is too large relative to $\lambda/\mu_{i,k}$, $\eta_{i,k}$ tends to take a value that is close to $\hat{\theta}_{i,k}$. At the $(k+1)$th step, $\hat{\theta}_{i,k+1}$ remains unchanged as $\hat{\theta}_{i,k}$, driving $\alpha_{i,k+1}$ to a large value, which further reduces $\lambda/\mu_{i,k+1}$. As a result, the ith variable will be discarded. Thus the selected terms should be those meeting convergence requirements and also contributing significantly to the prediction accuracy.

Suppose that M' variables remain with estimated coefficients $\hat{\theta}_i, i = 1, 2, \ldots, M'$ when the iteration converges. Some coefficients might be very small compared with others, i.e., $\hat{\theta}_i^2 \ll \|\hat{\boldsymbol{\theta}}\|_2^2$. In this case, those small weights lower than a threshold will be further pruned to obtain a compact model.

6.3.4 Coefficients estimation

The coefficients of the selected variables will be re-estimated. Let ST $= \{st_1, \ldots, st_{M''}\}$ denotes the set of indices of selected terms. These terms form a new candidate matrix $\mathbf{P}_{\mathrm{ST}} = [\boldsymbol{p}_{st_1}, \ldots, \boldsymbol{p}_{st_{M''}}] \in \mathbb{R}^{N \times M''}$. Related coefficients are $\boldsymbol{\theta}_{\mathrm{ST}} = [\theta_{st_1}, \ldots, \theta_{st_{M''}}]^T \in \mathbb{R}^{M''}$. Now the regression model becomes $\boldsymbol{y} = \mathbf{P}_{\mathrm{ST}}\boldsymbol{\theta}_{\mathrm{ST}} + \boldsymbol{e}$, where $\boldsymbol{y}$ denotes the model output and

e denotes the residual. Using the ordinary least squares estimator, the model coefficients are given by

$$\hat{\theta}_{\mathrm{ST}} = (\mathbf{P}_{\mathrm{ST}}^T \mathbf{P}_{\mathrm{ST}})^{-1} \mathbf{P}_{\mathrm{ST}}^T y \tag{6.23}$$

To improve the generalizability of the final model, ridge regression needs to be utilised. The model parameter estimation becomes

$$\hat{\theta}_{\mathrm{ST}} = (\mathbf{P}_{\mathrm{ST}}^T \mathbf{P}_{\mathrm{ST}} + \kappa \mathbf{I})^{-1} \mathbf{P}_{\mathrm{ST}}^T y \tag{6.24}$$

where κ is the penalty parameter.

Note that the identifiability of the estimation algorithm is guaranteed by the variable selection step. Firstly, the introduction of l_1-norm penalty term forces many of the model coefficients to zero, leaving only the important ones. This can be explained by the diamond-shaped constraint region in the space of coefficients. The corners represent situations in which other coefficients are exactly zero. Secondly, when multicollinearity exists in the candidate terms, the optimisation algorithm tends to choose one variable that provides unique information and set the parameters of others that have similar information to zero, which means the redundant variables no longer exist in $\mathbf{P}_{\mathrm{ST}}$ [215]. After the variable selection step, the matrix $\mathbf{P}_{\mathrm{ST}}$ has full column rank. Hence, the model coefficients can be uniquely determined by the least squares algorithm.

6.3.5 The implemented algorithm of LR-SALSA

The proposed approach consists of two stages, namely variable selection and coefficient estimation. To obtain a sparse model structure, local regularisation is introduced to solve the subproblem (6.5). In each iterative step, the evidence framework runs in a nested loop, estimating the coefficients θ and optimal local regularisation parameters μ. Within this loop, terms corresponding to large μ_i are removed. After that, variables ν, d and η are sequentially updated. The iteration stops when the prediction error converges. After the iteration, terms with small coefficients are pruned, resulting in a more compact model.

In the coefficient estimation stage, the coefficients are re-estimated using the system output and input of the selected terms. The complete procedure is outlined in **Algorithm 3**.

The complexity of the proposed method is discussed here. In each iteration of the LR-SALSA algorithm, solving the first subproblem (6.5) takes most of the computing time, and the other two subproblems (6.6) and (6.7) are solved directly. For (6.5), the update rules for

Algorithm 3 LR-SALSA

```
Variable selection stage:
Set $\boldsymbol{\alpha}_0, \beta_0, \lambda, tol_\alpha, \delta_\alpha, \delta_e$, and $\delta_\theta$; set $\boldsymbol{\nu}_\mathbf{0} = \boldsymbol{d}_\mathbf{0} = \boldsymbol{\eta}_\mathbf{0} = \mathbf{0}$
for $k = 1$ to $n$:
    Repeat
        Update $\boldsymbol{m}$ and $\mathbf{S}$ by (6.14) and (6.15)
        Update $\boldsymbol{\alpha}$ and $\beta$ by (6.16) and (6.17)
        if $|\alpha_i| \geq tol_\alpha$ then
            Term $\boldsymbol{p}_i$ is removed
        end if
    Until $\max\{\alpha_i^{last} - \alpha_i^{new}\} \leq \delta_\alpha$
    $\hat{\boldsymbol{\theta}}_k = \boldsymbol{m}, \boldsymbol{\mu} = \boldsymbol{\alpha}/\beta$
    $\boldsymbol{\nu}_k = \max(0, \hat{\boldsymbol{\theta}}_k + \boldsymbol{d}_{k-1} - \lambda/\boldsymbol{\mu}) - \max(0, -(\hat{\boldsymbol{\theta}}_k + \boldsymbol{d}_{k-1}) - \lambda/\boldsymbol{\mu})$
    $\boldsymbol{d}_k = \boldsymbol{d}_{k-1} + (\hat{\boldsymbol{\theta}}_k - \boldsymbol{\nu}_k)$
    $\boldsymbol{\eta}_k = \boldsymbol{\nu}_k - \boldsymbol{d}_k$
    $e_k = (\boldsymbol{y} - \mathbf{P}\hat{\boldsymbol{\theta}}_k)^T(\boldsymbol{y} - \mathbf{P}\hat{\boldsymbol{\theta}}_k)$
    if $|e_k - e_{k-1}| \leq \delta_e$ then
        break
    end if
end for
if $\hat{\theta}_i^2/\|\hat{\boldsymbol{\theta}}\|_2^2 \leq \delta_\theta$ then
    Term $\boldsymbol{p}_i$ is removed
end if
Determine the terms of the final model
Coefficient estimation stage:
Select the variables $\boldsymbol{y}$ and $\mathbf{P}_{\mathrm{ST}}$
Determine the coefficients by (6.24)
```

the hyperparameters $\boldsymbol{\alpha}$ and β depend on computing the posterior mean $\boldsymbol{m}$ and covariance matrix $\mathbf{S}$ of $\boldsymbol{\theta}$. The computation of $\boldsymbol{m}$ only involves matrix multiplications, whose amount is $O(M^2 + M^2 + MN)$. The computation of $\mathbf{S}$ requires an inverse operation of order $O(M^3)$ complexity and $O(M^2)$ memory storage. The computational complexity of $\boldsymbol{\alpha}$ and β is $O(M)$ and $O(MN)$, respectively. Therefore, the dominant cost in one iteration of hyperparameter updating should be $O(M^3)$.

In practice, the convergence is usually achieved in about tens of iterations. Even though there is a nested loop in the LR-SALSA algorithm, the solution can be obtained quickly as the pruning operation at each step rapidly reduces M to a very limited size. The computational efficiency improved by timely discarding redundant terms will be more significant when the initial value of M is very large. Based on the analysis above, the whole algorithm needs several times the cost of a standard SALSA algorithm, which is much less compared with the SAL method that needs about 25 times the cost of SALSA [206]. The complexity of the proposed algorithm is lower than the BAL method as well. In BAL, the computational complexity is $O(N^3)$ and the pruning procedure does not improve the efficiency [210].

6.3.6 The convergence

The convergence of the proposed LR-SALSA algorithm can be analysed under the framework of SALSA.

Theorem 1 [216] Consider problem (6.1), where f_1 and f_2 are closed, proper, convex functions. Consider arbitrary $\mu > 0$ and $\boldsymbol{v}_0, \boldsymbol{d}_0 \in \mathbb{R}^M$. Let $\{a_k \geq 0, k = 0, 1, \ldots, \infty\}$ and $\{b_k \geq 0, k = 0, 1, \ldots, \infty\}$ be two sequences such that

$$\sum_{k=0}^{\infty} a_k < \infty \quad \text{and} \quad \sum_{k=0}^{\infty} b_k < \infty \tag{6.25}$$

Consider three sequences $\{\hat{\boldsymbol{\theta}}_k \in \mathbb{R}^M, k = 0, 1, \ldots\}$, $\{\boldsymbol{v}_k \in \mathbb{R}^M, k = 0, 1, \ldots\}$, and $\{\boldsymbol{d}_k \in \mathbb{R}^M, k = 0, 1, \ldots\}$ that satisfy

$$a_k \geq \left\| \hat{\boldsymbol{\theta}}_{k+1} - \arg\min_{\boldsymbol{\theta}} \left\{ f_1(\boldsymbol{\theta}) + \frac{\mu}{2} \| \boldsymbol{\theta} - \boldsymbol{v}_k + \boldsymbol{d}_k \|_2^2 \right\} \right\| \tag{6.26}$$

$$b_k \geq \left\| \boldsymbol{v}_{k+1} - \arg\min_{\boldsymbol{v}} \left\{ f_2(\boldsymbol{v}) + \frac{\mu}{2} \| \hat{\boldsymbol{\theta}}_{k+1} - \boldsymbol{v} + \boldsymbol{d}_k \|_2^2 \right\} \right\| \tag{6.27}$$

$$\boldsymbol{d}_{k+1} = \boldsymbol{d}_k + (\hat{\boldsymbol{\theta}}_{k+1} - \boldsymbol{v}_{k+1}) \tag{6.28}$$

If (6.1) has a solution, the sequence $\{\hat{\boldsymbol{\theta}}_k\}$ converges, $\hat{\boldsymbol{\theta}}_k \longrightarrow \boldsymbol{\theta}^*$, where $\boldsymbol{\theta}^*$ is a solution of (6.1). However, if (6.1) does not have a solution, then at least one of the sequences $\{\boldsymbol{\nu}_k\}$ or $\{\boldsymbol{d}_k\}$ diverges.

This theorem shows that if the subproblems (6.5) and (6.6) can be solved exactly, or the sequence of errors is absolutely summable, convergence is guaranteed. As shown in [208], the objective function of (6.5) in SALSA is a strictly convex quadratic function, leading to the following solution:

$$\hat{\boldsymbol{\theta}}_{k+1} = (\mathbf{P}^T\mathbf{P} + \mu\mathbf{I})^{-1}(\mathbf{P}^T\boldsymbol{y} + \mu(\boldsymbol{\nu}_k - \boldsymbol{d}_k)) \tag{6.29}$$

which can be solved exactly in many cases. Because in a number of problems of interest, the term $(\mathbf{P}^T\mathbf{P} + \mu\mathbf{I})$ is invertible. For subproblem (6.6), it can be exactly solved by introducing the soft thresholding operator. Therefore, SALSA is guaranteed to converge.

Notice that (6.29) has a similar form to the maximum a posterior (MAP) estimate of $\boldsymbol{\theta}$, from observations $\boldsymbol{y} = \mathbf{P}\boldsymbol{\theta} + \boldsymbol{n}$, where $\boldsymbol{n}$ is white Gaussian noise of variance $1/\mu$ and $\boldsymbol{\theta}$ has a Gaussian prior $\mathcal{N}(\boldsymbol{\nu}_k - \boldsymbol{d}_k, \mathbf{I})$. In LR-SALSA, the local regularisation can apply individual penalty parameters on each element of $\boldsymbol{\theta}$. From Bayesian viewpoint, the Gaussian prior becomes $\mathcal{N}(\boldsymbol{\nu}_k - \boldsymbol{d}_k, \mathbf{A}^{-1})$, where $\mathbf{A} = \mathrm{diag}\{\alpha_1, \alpha_2, ..., \alpha_M\}$. In this case, the estimate of $\boldsymbol{\theta}$ is $\boldsymbol{m}$ at the convergence of $\boldsymbol{\alpha}$-optimisation, essentially taking the similar form to (6.29). Hence, the subproblem (6.5) in LR-SALSA can be exactly solved as long as $\boldsymbol{\alpha}$-optimisation converges.

As shown in the Appendix, an implicit solution for $\boldsymbol{\alpha}$ is derived by directly maximising the marginal likelihood function $\ln p(\boldsymbol{y}|\mathbf{P})$. The re-estimation procedure will converge at a point where $\ln p(\boldsymbol{y}|\mathbf{P})$ reaches a stationary point. During the re-estimation, many of the α_i tend to infinity, implying that $p(\theta_i|\boldsymbol{y}, \boldsymbol{\alpha}, \beta)$ becomes highly peaked at η_i [214]. Then these α_i are discarded from the next iteration and the corresponding terms can be pruned. As illustrated in [211], at the optimum value of $\boldsymbol{\alpha}$, the effective total number of well-determined parameters is also ensured. Therefore, the whole LR-SALSA algorithm is guaranteed to converge.

6.4 Simulation study

In this section, two nonlinear examples represented by Nonlinear AutoRegressive with eXogenous inputs (NARX) models are employed to evaluate the sparsity, accuracy, and computation speed of the feature selection methods. A systematic introduction to NARX models can be found in [217], which will not be repeated here. The utilisation of the NARX model for this simulation study is based on the assumption that the accurate selection of NARX model terms and accurate prediction of model parameters can be used as a reliable indicator of the capability of the feature selection method in generating representative features and making accurate predictions.

For this simulation study, except for LR-SALSA, many algorithms including OFR [205], LASSO [207], SAL [206], SBL [209], and BAL [210] are implemented for the sake of comprehensive comparison. To ensure a fair evaluation, each algorithm has been executed multiple times with different random seeds. Both the best result and statistical analysis of performance indices after repeated executions are presented in this section.

6.4.1 Example 1

First, consider the benchmark nonlinear system [206]:

$$\begin{aligned} z(t) =& -0.5z(t-2) + 0.7z(t-1)u(t-1) + 0.6u^2(t-2) + \\ & 0.2z^3(t-1) - 0.7z(t-2)u^2(t-2) \\ y(t) =& z(t) + e(t) \end{aligned} \tag{6.30}$$

where $u(t)$ and $z(t)$ denote the system input and output at time t, respectively. $u(t)$ is a uniformly distributed white noise within the range of $[-1, 1]$. $z(t)$ is the output of the system that is excited by $u(t)$. An external Gaussian noise sequence $e(t)$ is added to $z(t)$. The signal-to-noise ratio (SNR) is set to 15 dB. In this simulation study, the maximum lags for input and output are selected as 3 and 4, respectively. The linear candidate terms are $z(t-1), z(t-2), z(t-3), z(t-4), u(t-1), u(t-2)$ and $u(t-3)$. In terms of nonlinearity, the maximum polynomial degree is selected as 3. As a result, based on the dictionary generation equation (6.10), the total number of candidate terms is 119. In this example, 3000 input and output samples are generated for model learning. Thus, the dimension of matrix $\mathbf{P}$ is 3000×119. The mean absolute error (MAE) of the estimated coefficients is used to evaluate the model performance

$$\mathrm{MAE}_\theta = \frac{1}{M}\sum_{i=1}^{M} |\theta_i - \hat{\theta}_i| \tag{6.31}$$

In the implementation of the proposed LR-SALSA algorithm, some parameters need to be determined in advance. The tolerance tol_α and stop criterion δ_α used in local regularisation parameter learning were set to $1e^5$ and $1e^{-5}$, respectively, according to the suggestions in [214]. The stopping criteria δ_e and δ_θ for variable selection were $1e^{-5}$ and 0.01, which were reasonably small [206]. The l_1-norm parameter λ was set to 1. In the OFR algorithm, the important terms were selected according to the error reduction ratio criterion, which was

Table 6.1: The simulation results of example 1

Terms	Coefficient	**LR-SALSA**	OFR	LASSO	SAL	SBL	BAL
$z(t-2)$	-0.5	-0.498	-0.5014	-0.4967	-0.5021	-0.4981	-0.5002
$z(t-1)u(t-1)$	0.7	0.6954	0.698	0.6866	0.6901	0.7039	0.7024
$u^2(t-2)$	0.6	0.5992	0.5997	0.5898	0.5967	0.6018	0.5994
$z^3(t-1)$	0.2	0.2008	0.2018	0.1911	0.2005	0.1986	0.1976
$z(t-2)u^2(t-2)$	-0.7	-0.7064	-0.6992	-0.6575	-0.6884	-0.6952	-0.7009
$z(t-4)$	-	-	-	0.0011	-	-	-
$u^2(t-1)$	-	-	-	0.0008	-	-	-
$u(t-1)u^2(t-3)$	-	-	-	0.002	-	-	-
$z(t-4)u^2(t-2)$	-	-	-0.0022	-	-	-	-
MAE$_\theta$	-	**0.0029**	0.0012	0.0157	0.0055	0.0028	0.0013

carefully tuned and set to 0.05 in this study [210]. In other algorithms, the l_1-norm parameter λ was critical to model performance. The cross-validation method was adopted to select a value that results in the smallest standard error [218]. The selected parameters for these methods were: $\lambda_{\text{LASSO}} = 7.2e^{-4}$, $\lambda_{\text{SAL}} = 1$, $\lambda_{\text{SBL}} = 0.4$, and $\lambda_{\text{BAL}} = 0.1$. Additionally, in SAL, the subsampling was repeated 10 times and $\delta_{thr} = 0.7$. In SBL, the CVX solver was used to address the optimisation problem and the number of repeated steps was 5 [209].

Table 6.1 presents the best results obtained in this case. It can be seen that all methods can produce sparse models. OFR estimates the model coefficients with the highest accuracy, but a redundant term $z(t-4)u^2(t-2)$ is also selected. LASSO selects more redundant terms and the coefficients of the correct terms are less accurate than others. The other four methods accurately choose the correct terms and estimate coefficients at a similar level of precision. Among them, BAL stands out with a MAE$_\theta$ of 0.0013.

To evaluate the stability of these methods,100 simulation executions have been carried out using different random seeds and recorded the resulting performance indices. Fig. 6.1 shows the box plots for the number of selected terms, MAE$_\theta$, and computation time. As can be seen in Fig. 6.1 (a), the LR-SALSA and SBL algorithms consistently identify the correct terms. The SAL and BAL methods choose redundant terms occasionally. In contrast, the OFR and LASSO algorithms tend to select additional terms. In terms of estimation accuracy, as shown in Fig. 6.1 (b), OFR has the lowest and highest MAE$_\theta$, showing worse stability. The proposed LR-SALSA algorithm outperforms LASSO and SAL and achieves similar estimation accuracy to SBL and BAL. However, as shown in Fig. 6.1 (c), LR-SALSA takes the least amount of time to construct a model compared with other methods except for OFR. It takes an extremely long time for feature selection using the SBL algorithm, which indicates the limitation of using the third-party solver. As a result, compared with the other methods, the proposed LR-SALSA method offers a superior accuracy in generating sparse models meanwhile ensuring fast computation.

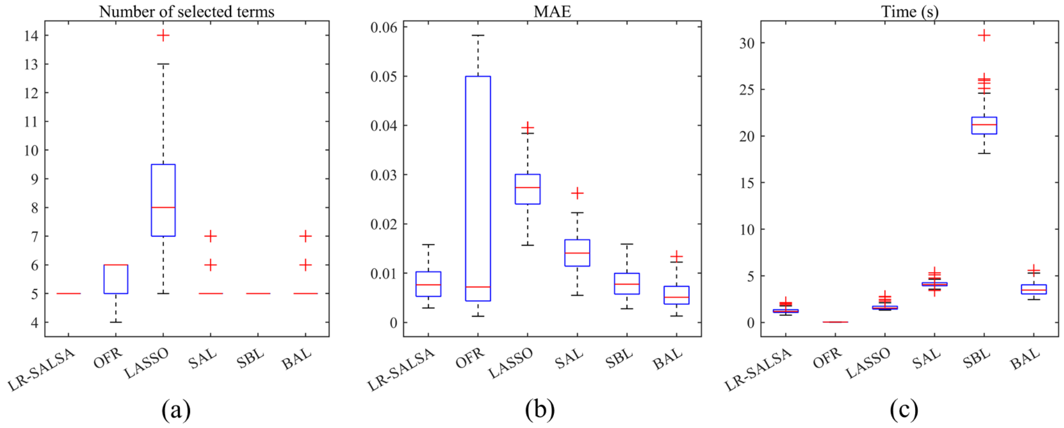


Figure 6.1: Box plots of the results produced by LR-SALSA and other common algorithms in example 1: (a) number of selected terms, (b) MAE$_\theta$, and (c) computation time.

6.4.2 Example 2

Consider another nonlinear system [206]:

$$\begin{aligned} z(t) =& u(t-1) - 0.3u(t-2) - 0.4u(t-3) + 0.25u(t-1)u(t-2) - \\ & 0.2u(t-2)u(t-3) - 0.3u^3(t-1) + 0.24u^3(t-2) + 0.8z(t-1) \\ y(t) =& z(t) + e(t) \end{aligned} \tag{6.32}$$

where $u(t)$ denotes the system input and $z(t)$ denotes the output at time t. A white noise $u(t) \in [-1, 1]$ sampled from the uniform distribution is used as the system input. For the Gaussian noise sequence $e(t)$, the SNR is set to 15 dB. In this example, the maximum lags for input and output are selected as 2 and 3. The maximum polynomial degree is selected as 3. Therefore, these delayed inputs and outputs can form a model term candidate pool with 55 linear and nonlinear terms. In total, 3500 samples are generated. The size of $\mathbf{P}$ is 3500×55. The MAE$_\theta$ is also used to test the model performance.

The parameter settings for the proposed algorithm were consistent with those in Example 1. Nevertheless, key parameters used in the comparative methods have changed. In the OFR algorithm, the stopping criterion was set to 0.033. In the other four algorithms, the l_1-norm parameters were selected as the following values: $\lambda_{\text{LASSO}} = 3.5e^{-4}$, $\lambda_{\text{SAL}} = 0.01$, $\lambda_{\text{SBL}} = 0.03$, and $\lambda_{\text{BAL}} = 0.05$. The best simulation results are presented in Table 6.2. Similar to the results in Example 1, the LR-SALSA, SBL, and BAL algorithms can estimate the coefficients of correct terms more accurately. Even though SBL yields the lowest MAE$_\theta$ of 0.0044, 6 redundant terms are included in the final model, making the result unreliable. LASSO also produces a redundant model comprising 14 terms and neglects the essential term $u(t-2)$. The OFR and SAL algorithms have selected the true terms but have not estimated the coefficients as accurately as the proposed LR-SALSA algorithm.

Table 6.2: The simulation results of example 2

Terms	Coefficient	**LR-SALSA**	OFR	LASSO	SAL	SBL	BAL
$z(t-1)$	0.8	0.7917	0.8959	0.5157	0.7475	0.8045	0.7847
$u(t-1)$	1	0.9942	0.9942	0.9769	1.009	0.9978	1.0038
$u(t-2)$	-0.3	-0.2808	-0.3871	-	-0.2581	-0.2995	-0.2835
$u(t-3)$	-0.4	-0.3977	-0.4456	-0.2477	-0.3733	-0.459	-0.3983
$u(t-2)u(t-2)$	0.25	0.2484	0.2557	0.2473	0.2401	0.2565	0.249
$u(t-2)u(t-3)$	-0.2	-0.2035	-0.2269	-0.1238	-0.1848	-0.2056	-0.2025
$u^3(t-1)$	-0.3	-0.2918	-0.2917	-0.2699	-0.3101	-0.3002	-0.3155
$u^3(t-2)$	0.24	0.2332	0.2571	0.1301	0.2453	0.2301	0.2316
$z^2(t-1)$	-	-	-	-0.0061	-	-	-
$z(t-1)u(t-1)$	-	-	-	0.0057	-	-	-
$z(t-2)u(t-3)$	-	-	-	0.0122	-	-	-
$z^3(t-2)$	-	-	-	-0.0107	-	-	-
$u^2(t-1)u(t-3)$	-	-	-	-0.0017	-	-	-
$u(t-1)u^2(t-2)$	-	-	-	-0.0091	-	-	-
$u^2(t-2)u(t-3)$	-	-	-	-0.0003	-	-	-
$z(t-1)u(t-3)$	-	-	-	-	-	0.0041	-
$u(t-1)u(t-3)$	-	-	-	-	-	0.0058	-
$z(t-1)u(t-1)u(t-2)$	-	-	-	-	-	-0.0051	-
$z(t-1)u(t-1)u(t-3)$	-	-	-	-	-	-0.0033	-
$z(t-2)u^2(t-2)$	-	-	-	-	-	0.0053	-
$u(t-2)u^2(t-3)$	-	-	-	-	-	-0.102	-
MAE$_\theta$	-	**0.0070**	0.0366	0.1223	0.0213	0.0044	0.0080

The stability test has been conducted in this example as well, and Fig. 6.2 shows the box plots resulting from 100 simulation executions with different random seeds. As can be seen in Fig. 6.2 (a), the average numbers of selected terms of LR-SALSA and BAL are approximately 8, aligning with the true model term number in Eq. (6.32). The OFR, LASSO, and SAL algorithms tend to estimate an over-sparse model, consistently selecting fewer than 6 terms. This indicates that some important terms may be ignored. On the contrary, SBL usually yields a model with redundant terms. Fig. 6.3 shows the occurrences of true terms in 100 executions. It can be found that terms like $u(t-1)$, $u(t-1)u(t-2)$, and $u^3(t-1)$ are consistently chosen by all six algorithms nearly in every execution. However, other important terms are selected in a higher occurrence only by using LR-SALSA, SBL, and BAL methods. Compared with SBL and BAL, LR-SALSA exhibits superior estimation accuracy and efficiency as can be seen in Figs. 6.2 (b) and (c). Furthermore, in this example, even though OFR and LASSO take less computation time but also produce higher estimation errors. The results in both Examples 1 and 2 demonstrate that LR-SALSA is able to build compact and accurate models, and significantly reduce the computational burden at the same time.

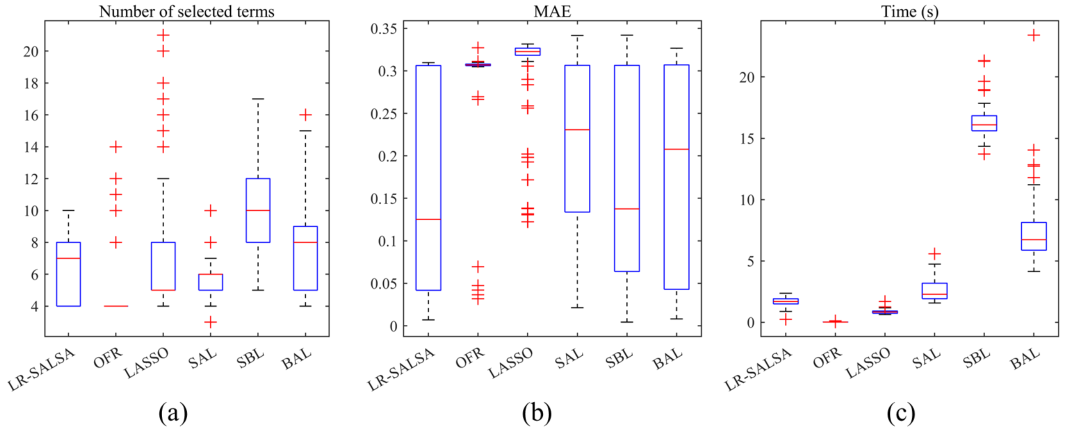


Figure 6.2: Box plots of the results produced by LR-SALSA and other common algorithms in example 2: (a) number of selected terms, (b) MAE$_\theta$, and (c) computation time.

6.5 Experimental study

To validate the effectiveness of the proposed feature selection method in real-world applications, the datasets obtained from face milling and dynamic milling experiments were employed to develop a model for tool wear prediction. Similar to the simulation study, as a comparison, other common l_1-norm-based feature selection methods were applied to select important signal features. The model performance was evaluated in terms of sparsity, accuracy, and efficiency.

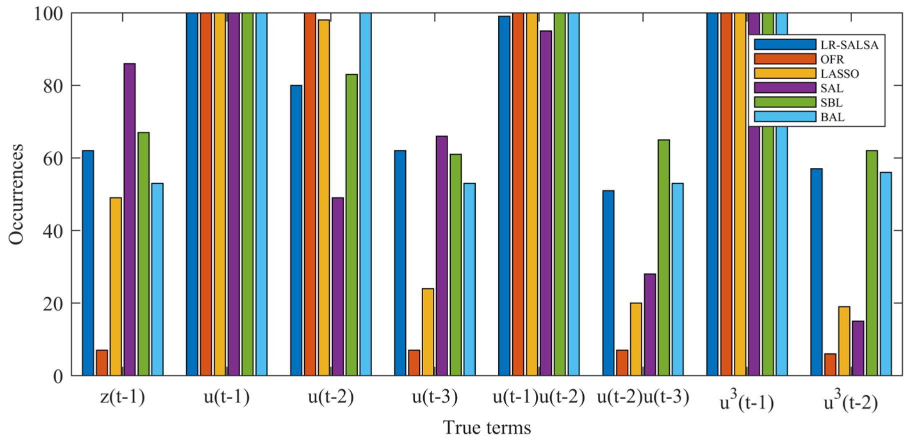


Figure 6.3: The occurrences of all true terms in 100 executions with different random seeds in example 2.

6.5.1 Experiment 1: face milling process

Experimental setup

The dataset applied to this case study is provided by the 2010 Prognostics and Health Management Data Challenge [115]. Fig. 6.4 (a) shows the experimental setup where the milling process was conducted on a high-speed milling machine (Roders Tech RFM 760). The surface of an Inconel 718 workpiece was machined layer by layer using a three-flute ball end mill with a slope of 60°. The machining parameters were set as follows: the spindle speed of the cutter was 10400 RPM; feed rate was 1555 mm/min; radial cutting depth was 0.125 mm; axial cutting depth was 0.2 mm. During the milling process, tri-axial force, tri-axial vibration, and AE signals were collected using dynamometers, accelerometers, and AE sensors respectively. The sampling frequency was set to 50 kHz. The flank wear of each flute was measured using a microscope after each cut. Figure 6.4 (b) presents the tool wear measurements for three cutters, namely C1, C4, and C6. The dataset of one cutter comprises 315 samples corresponding to 315 cuts. It is worth mentioning that the datasets from C2, C3 and C5 were not used in this study because the tool wear measurements were not available for these three cutters.

Algorithm implementation

In this experiment, six channels of signals, including tri-axial force and vibration, were utilised to construct a tool wear prediction model. The AE signal was not used because the dominant frequency range was too high compared with the tooth passing frequency. A total of 13 time and frequency domain features, as listed in Table 6.3, were extracted from each signal. Thus

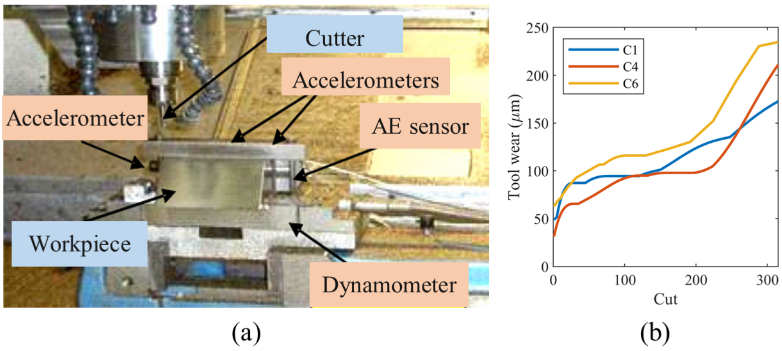


Figure 6.4: (a) Experimental setup [115], (b) the measurements of tool wear.

Table 6.3: List of the extracted features in experiment 1

Features	Time domain	mean, variance, skewness, kurtosis, RMS, peak-to-peak value, crest factor
	Frequency domain	mean, variance, skewness, kurtosis, peak-to-peak value, harmonics

the number of candidate features was 78 (= 13×6) for each sample. The datasets from any two cutters among C1, C4, and C6, comprising 630 samples, were allocated for training and validation. Among them, 90% (567 samples) were randomly designated as the training set, and the remaining 10% (63 samples) were for validation purpose. The dataset from the other cutter (315 samples) was used for testing. Considering the features were directly used as inputs, the dimensions of the training, validation, and testing feature matrices were 567×78, 63×78, and 315×78, respectively.

After the feature dictionary was generated, each column was normalised using the training set, ensuring that all variables were adjusted to the same scale. Subsequently, the feature selection algorithms were applied to build sparse regression models for tool wear prediction. The prediction accuracy was evaluated on the testing set using two indices: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), which are defined below,

$$\begin{aligned} \text{MAE} &= \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \\ \text{RMSE} &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \end{aligned} \tag{6.33}$$

where y_i and $\hat{y}_i$ denote the true and predicted tool wear of the ith cutting pass, respectively.

Here the tuning of key parameters is introduced briefly. Most of the parameters in LR-

SALSA shared the same values as the simulation study, except that the iteration stopping criterion δ_e was changed to $1e^{-3}$ considering the range of target values and the parameter λ was selected by cross-validation. It should be noted that in the second stage of LR-SALSA, ridge regression with a penalty coefficient of 2 was employed to estimate model coefficients to further avoid overfitting. In the SAL method, the number of subsampling rounds was 10 and δ_{thr} was 0.6. In SBL, the number of total steps was 10. Additionally, cross-validation was utilised to determine the error reduction ratio in OFR and the regularisation parameter λ in other methods.

Experimental results

Fig.6.5 shows the tool wear prediction results for all cutters using different feature selection methods. The number of selected features, prediction accuracy, and execution time are summarised in Table 6.4. It should be noted that all the tests were conducted on a laptop with an Intel(R) Core(TM) i7-7700HQ CPU and 16GB memory. As can be observed, the proposed LR-SALSA selected 8, 7, and 4 features for C1, C4, and C6. While the number of features selected by LASSO varies from 19 to 49. The other methods produced sparse models with 4 to 14 features. This indicates that LR-SALSA can achieve better model sparsity.

From the prediction results, it can be found LR-SALSA method shows better fitting performance, especially for C1 and C6. The other methods can also make accurate predictions in several cases, for example, the result of C6 using SBL. MAE and RMSE can quantify the prediction accuracy. In the LR-SALSA method, these two indices vary from 8.38 to 10.58 and from 9.97 to 13.92, respectively. However, the other methods yield larger MAE and RMSE values. Even though the predicted result of SBL fits the actual tool wear well, the indices are twice as large as the proposed method.

In terms of computational efficiency, OFR takes the least time of less than 1 second in all three cases. LR-SALSA takes about 2 seconds, which is close to LASSO and BAL. While SAL and SBL take the longest time due to the use of subsampling and the third-party solver. Thus the conclusion obtained in the simulation study that LR-SALSA has a computational advantage over SAL, SBL, and BAL still holds. In this experimental study, the results demonstrate the benefits of LR-SALSA in selecting less but representative features and ensuring fast computation with high prediction accuracy.

Table 6.4: The testing results of C1, C4, and C6 in experiment 1 (the maximum polynomial degree: 1)

Algorithms		**LR-SALSA**	OFR	LASSO	SAL	SBL	BAL
Number of features	C1	8	14	19	8	8	12
	C4	7	8	49	10	14	5
	C6	4	4	45	11	12	8
MAE	C1	8.52	20.95	24.02	27.35	16.77	18.41
	C4	10.58	19.18	17.38	16.92	18.35	14.41
	C6	8.38	31.14	48.82	24.13	16.17	23.5
RMSE	C1	11.27	24.68	28.98	39.15	18.7	20.44
	C4	13.92	21.16	20.05	18.65	20.73	19.34
	C6	9.97	35.26	50.33	25.7	17.57	26.69
Execution time (sec)	C1	2.21	0.34	2.52	6.82	6.41	3.4
	C4	1.97	0.21	2.65	6.56	6.07	3.81
	C6	2.33	0.13	2.93	7.11	5.7	3.34

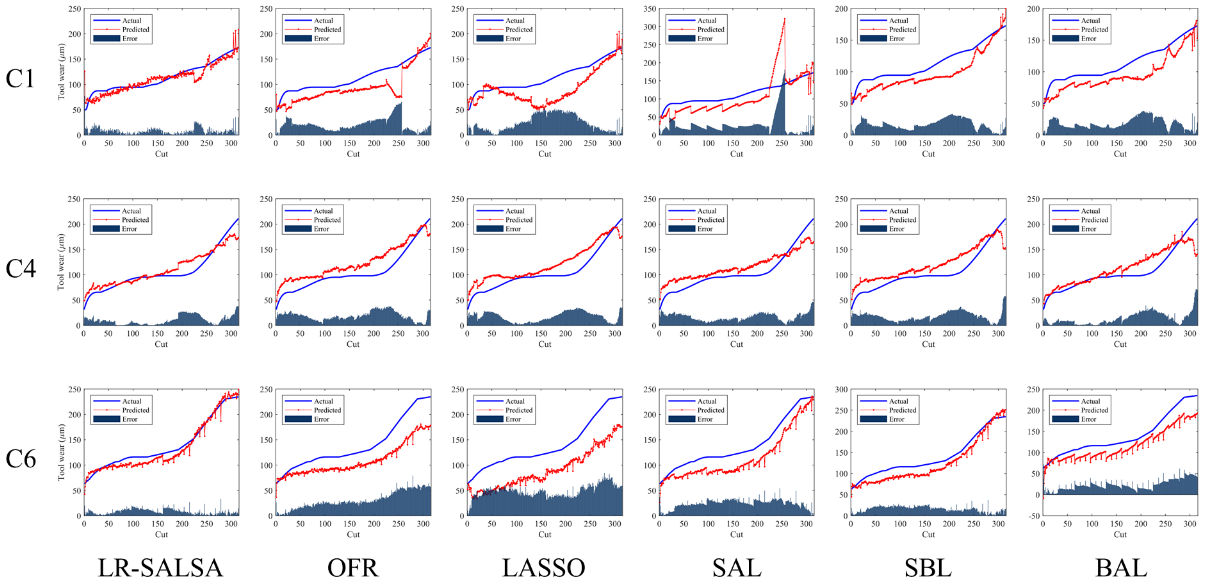


Figure 6.5: Tool wear prediction results in experiment 1, from top to bottom: for cutters C1, C4 and C6; from left to right: using LR-SALSA, OFR, LASSO, SAL, SBL, and BAL methods. (the maximum polynomial degree: 1)

The influence of maximum polynomial degree

The construction of dictionary matrix $\mathbf{P}$ plays an important role in the feature selection problem. In the simulation study, the maximum lags and polynomial degrees are selected directly, which are not known in most cases. The experimental results mentioned above are from 78 linear terms of raw signal features. However, the polynomial expansion can enrich the extracted information, which may improve the model accuracy [217]. Hence, it is necessary to investigate the performance of the feature selection methods with more complicated inputs. The second-degree terms are generated by combining different features (for instance, X-vibration Mean×Y-vibration Variance), leading to 3159 ($= 78 + 78 \times 79 \div 2$) candidate terms in total. After the data splitting, the dimensions of the training, validation, and testing feature matrices become 567×3159, 63×3159, and 315×3159, respectively.

All the feature selection methods are applied to the new matrices. The parameters are determined in the same way as aforementioned. Table 6.5 shows the testing performance for all cutters. It can be found that, except for SAL, most methods tend to select more important features when the second-degree terms are included in the dictionary matrix. However, the feature selection still works as the remaining features are significantly less than the candidates. Contrary to expectations, the prediction accuracy decreases compared with Table 6.4 in most cases. While the proposed LR-SALSA always achieves the lowest fitting errors. Its advantage becomes more obvious given the fact SAL, SBL, and BAL take dozens of times longer in feature selection. However, OFR still completes the task rapidly owing to the characteristic of the forward selection algorithm.

This case study further demonstrates that the proposed LR-SALSA is able to achieve a good trade-off in model sparsity, accuracy and computational efficiency. Its high execution speed is more significant in many scenarios involving complicated datasets. However, it should be acknowledged that sometimes more candidate terms do not necessarily bring about more accurate models. Too many redundant features make it difficult to identify the truly important features. Therefore, how to select an appropriate polynomial degree to construct the feature dictionary deserves more study.

6.5.2 Experiment 2: dynamic milling process

Nowadays, dynamic milling has become more popular in metal machining. Accurately predicting the tool wear in dynamic milling is quite challenging due to the complicated interaction between the tool and workpiece. In this experiment, the performance of the proposed method

Table 6.5: The testing results of C1, C4, and C6 in experiment 1 (the maximum polynomial degree: 2)

Algorithms		**LR-SALSA**	OFR	LASSO	SAL	SBL	BAL
Number of features	C1	12	16	116	8	34	5
	C4	13	19	63	4	15	15
	C6	9	3	136	7	9	5
MAE	C1	13.71	19.64	17.03	14.5	19.31	16.1
	C4	12.19	20.23	20.49	25.97	15.4	14.48
	C6	13.1	33.51	89.29	28.92	22.77	20.18
RMSE	C1	17.23	24.65	19.9	18.16	27.01	20.2
	C4	14.38	22.07	22.97	34.31	19.01	17.7
	C6	16.36	36.69	97.9	30.56	23.93	21.84
Execution time (sec)	C1	10.31	0.92	45.41	472.72	286.07	309.95
	C4	10.58	1.23	44.89	623.94	130.64	216.98
	C6	9.51	0.07	18.2	1267.74	57.58	212.75

has been investigated in dynamic milling. The dataset was from the same experiment as Case 2 in Chapter 5. The experimental setup will not be repeated here. During the experiment, the flank wear of the cutting tool was measured by a microscope after each cut. Figure 6.6 shows the measurements of tool wear. All three cutting tools (C1, C2, and C3) completed the milling of six layers. The flank wear of each tool reached 395.3 μm, 421.1 μm and 490.0 μm finally.

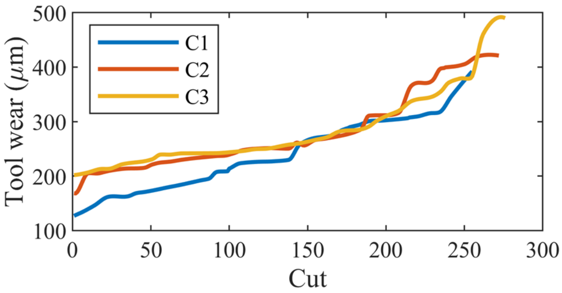


Figure 6.6: The measurement of tool wear in experiment 2.

The raw datasets in this case consisted of nine variables containing force and vibration. The number of available samples of the three cutters was 252, 272, and 276, respectively. Similar to the design of Experiment 1, in this experiment, the datasets from any two cutters

Table 6.6: The testing results of C1, C2, and C3 in experiment 2

Algorithms		**LR-SALSA**	LASSO	SAL
Number of terms	C1	3	47	3
	C2	5	54	5
	C3	1	17	2
MAE	C1	25.09	114.5	67.88
	C2	35.16	66.34	69.54
	C3	19.58	331.6	156.5
RMSE	C1	32.14	166	76.8
	C2	44.86	71.51	78.43
	C3	22.24	407.1	163.5

were used for training and validation and the dataset from the other cutter for testing. In terms of feature extraction, only the time domain features listed in Table 6.3 were extracted. The second-degree polynomial combination was exploited to enlarge the size of the pool of candidates, reaching up to 2079.

In this case, only the LASSO and SAL methods are selected in the comparative study because they could represent the properties of traditional and novel l_1-norm-based feature selection methods as shown in the simulation study. MAE and RMSE were used to quantify the model's fitting performance. The values of key parameters were defined in the same way as in experiment 1. In LR-SALSA, rigid regression was also adopted in coefficient estimation.

Figure 6.7 shows the tool wear prediction results during the testing stage. Obviously, the achieved prediction accuracy is not as good as in Experiment 1 because the dynamic milling strategy is utilised. However, the predicted curves of LR-SALSA show the best fit to the true wear curves. For LASSO and SAL, in most cases, the learned model can only capture a general increasing trend of tool wear, which might result from inappropriate model coefficients estimation rather than term selection. However, if the predicted tool wear severely deviates from the true value, the selected terms should not have good correlation with the tool wear. Table 6.6 summarises the performance indices in this experiment. It demonstrates again the necessity of selecting less but important terms rather than redundant terms in tool wear prediction. The proposed method achieves the best estimation accuracy.

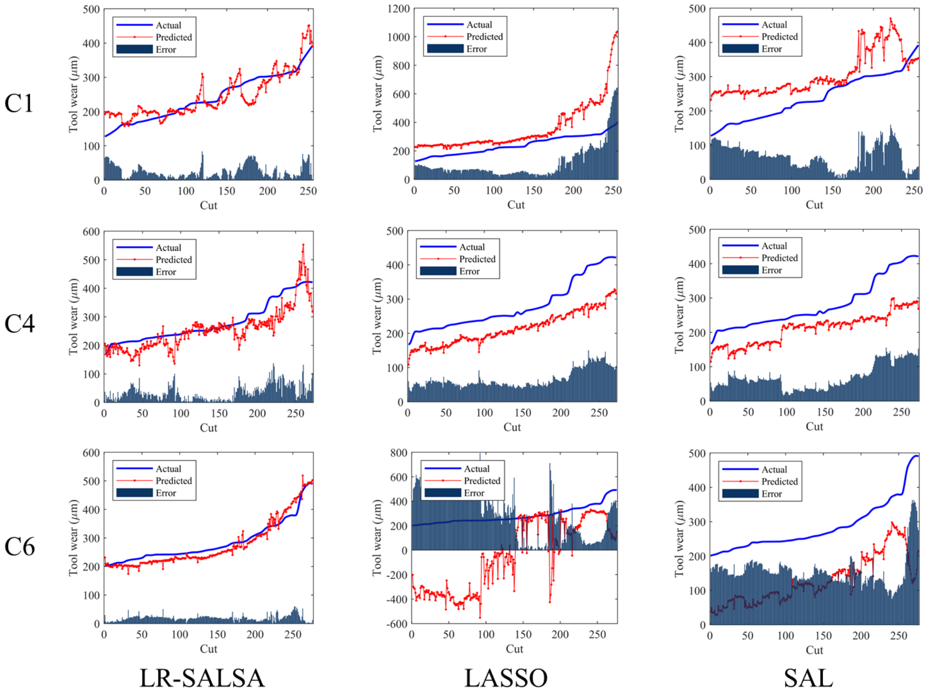


Figure 6.7: Tool wear prediction results in experiment 2, from top to bottom: C1, C2, and C3; from left to right: using LR-SALSA, LASSO, and SAL.

6.6 Discussion

To better explain the advantages of the proposed LR-SALSA-based feature selection approach in condition monitoring, some discussions are summarised as follows.

- *Scope of application*: The proposed algorithm can be applied to obtain a linear-in-the-parameter model, where the relationship between the response y and the parameters $\boldsymbol{\theta}$ is linear. However, a model being linear-in-the-parameter may also represent a nonlinear relationship between the input and output variables. Because the input variables can be transformed or combined in non-linear ways to construct the dictionary matrix $\mathbf{P}$, for example, the polynomial-expansion models (6.30) and (6.32). Linear-in-the-parameter models are widely applied across various fields due to their simplicity, interpretability, and computational efficiency. As the original set of available terms may be very large, feature selection is a crucial step when building a linear-in-the-parameter model. The proposed LR-SALSA algorithm can be applied to both linear and nonlinear models by choosing a subset of important terms and producing a sparse model.

- *Sparsity and accuracy*: From both simulation and experimental studies, it is observed that conventional OFR and LASSO methods often select over-sparse or redundant features, reducing the accuracy of model predictions. SAL and SBL methods have achieved better estimation accuracy but cannot always choose the right terms. Only LR-SALSA and BAL can build sparse models accurately. The unique contribution of this work is that the proposed LR-SALSA algorithm can automatically select the most representative features, which originates from a straightforward local regularisation strategy. The regularisation parameters are directly related to the contribution of candidate features. As a result, the proposed method can provide a more compact model, addressing the difficulties of over-sparsity and redundancy.

- *Computational speed*: It can be found from the simulation and experimental studies that, the proposed LR-SALSA method exhibits higher computational efficiency when compared with the modern SAL, SBL, and BAL methods. The increased efficiency of LR-SALSA is attributed to its design, which conducts the SALSA algorithm only once compared with SAL, avoids the use of third-party solvers compared with SBL, and requires less computation burden compared with BAL.

- *Limitations*: The number of tuning parameters of LR-SALSA, at this stage, is greater than some of the established methods. Their settings have been introduced in both simulation and experimental studies, which are summarised as follows.

tol_α: it is used as a threshold to determine which terms should be removed in the learning of local regularisation parameters. Increasing tol_α will increase the execution time. It should be reasonably large and is fixed as $1e^5$ in this study according to [214].

δ_α: it is used to determine when the learning of local regularisation parameters converges. Increasing δ_α will reduce the execution time. It should be reasonably small and is fixed as $1e^{-5}$ in this study according to [214].

δ_e: it is used to determine when the whole algorithm converges. Increasing δ_e will reduce the execution time and increase the number of selected features. It should be determined according to the range of target values and is set to $1e^{-5}$ and $1e^{-3}$ in simulation and experiment study, respectively.

δ_θ: it is used as a threshold to determine which terms should be further removed after the iteration process. Increasing δ_θ will reduce the number of selected features. It is set to 0.01 as used in [206].

λ: it is the l_1-norm regularisation parameter. Increasing λ usually results in more selected features. It is determined automatically through cross-validation.

From the analysis above, it is found that most parameters can be determined according to empirical knowledge. However, in specific applications, fine-tuning of the parameters is unavoidable. A systematic analysis is needed to guide the setting of key parameters and enhance the applicability of LR-SALSA. This will be a focal point of future research endeavours.

6.7 Conclusion

Due to the benefits of avoiding overfitting and enhancing computational efficiency, feature selection becomes a research hotspot in data-driven process states prediction. This chapter introduces the local regularisation assisted split augmented lagrangian shrinkage algorithm as a solution to the feature selection problem. The objective is to solve a l_1-norm optimisation problem where the problem is decomposed into three subproblems that can be solved separately. In the first subproblem, individual penalty parameters are introduced to each coefficient such that redundant terms are removed efficiently. An iterative algorithm, based on the Bayesian evidence framework, is derived to optimise the local regularisation hyperparameters automatically and guarantee model sparsity. After the important terms are determined, model coefficients are re-estimated by the least squares method. Consequently, the feature selection problem is resolved in a single SALSA iteration, significantly reducing computational expenses. The complexity and convergence of the proposed algorithm are presented. The effectiveness of the proposed approach is demonstrated through both simulation and experimental studies. The results show that only LR-SALSA has a good trade-off among model sparsity, estimation accuracy, and computational efficiency in comparison to five existing feature selection methods. The proposed method is expected to be applicable in process monitoring of many industrial systems.

6.8 Appendix

The evidence framework

In the Bayesian evidence framework, the following prior distributions of $\boldsymbol{y}$ and $\boldsymbol{\theta}$ are first introduced

$$p(\boldsymbol{y}|\mathbf{P},\boldsymbol{\theta},\beta) = \mathcal{N}\left(\boldsymbol{y}|\mathbf{P}\boldsymbol{\theta},\beta^{-1}\mathbf{I}\right) \tag{6.34}$$

$$p(\boldsymbol{\theta}|\boldsymbol{\eta},\boldsymbol{\alpha}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\eta},\mathbf{A}^{-1}) \tag{6.35}$$

where β and α_i govern the precision parameters of noise and θ_i, and $\mathbf{A} = \mathrm{diag}\{\alpha_1,\alpha_2,...,\alpha_M\}$ denotes the precision matrix.

Since $p(\boldsymbol{\theta}|\boldsymbol{y},\mathbf{P},\boldsymbol{\eta},\boldsymbol{\alpha},\beta) \propto p(\boldsymbol{y}|\mathbf{P},\boldsymbol{\theta},\beta)p(\boldsymbol{\theta}|\boldsymbol{\eta},\boldsymbol{\alpha})$, the posterior distribution for $\boldsymbol{\theta}$ is Gaussian and takes the form

$$p(\boldsymbol{\theta}|\boldsymbol{y},\mathbf{P},\boldsymbol{\eta},\boldsymbol{\alpha},\beta) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{m},\mathbf{S}) \tag{6.36}$$

where the mean and covariance matrix are given by

$$\boldsymbol{m} = \mathbf{S}(\mathbf{A}\boldsymbol{\eta} + \beta\mathbf{P}^T\boldsymbol{y}) \tag{6.37}$$

$$\mathbf{S} = (\mathbf{A} + \beta\mathbf{P}^T\mathbf{P})^{-1} \tag{6.38}$$

The unknown parameters $\boldsymbol{\alpha}$ and β are determined by evaluating the evidence function [214]. The target is to maximise the marginal likelihood function $p(\boldsymbol{y}|\mathbf{P},\boldsymbol{\eta},\boldsymbol{\alpha},\beta)$ by integrating over the weight vector $\boldsymbol{\theta}$

$$\begin{aligned} p(\boldsymbol{y}|\mathbf{P},\boldsymbol{\eta},\boldsymbol{\alpha},\beta) &= \int p(\boldsymbol{y}|\mathbf{P},\boldsymbol{\theta},\beta)p(\boldsymbol{\theta}|\boldsymbol{\eta},\boldsymbol{\alpha})\,d\boldsymbol{\theta} \\ &= \left(\frac{\beta}{2\pi}\right)^{\frac{N}{2}}\left(\frac{1}{2\pi}\right)^{\frac{M}{2}}\prod_{i=1}^{M}\alpha_i^{\frac{1}{2}}\int \exp\{-E(\boldsymbol{\theta})\}\,d\boldsymbol{\theta} \end{aligned} \tag{6.39}$$

where

$$E(\boldsymbol{\theta}) =\frac{\beta}{2}\boldsymbol{e}^T\boldsymbol{e} + \frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\eta})^T\mathbf{A}(\boldsymbol{\theta}-\boldsymbol{\eta}) = E(\boldsymbol{y},\boldsymbol{\eta}) + \frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{m})^T\mathbf{S}^{-1}(\boldsymbol{\theta}-\boldsymbol{m}) \tag{6.40}$$

$$E(\boldsymbol{y},\boldsymbol{\eta}) = \frac{1}{2}(\beta\boldsymbol{y}^T\boldsymbol{y} + \boldsymbol{\eta}^T\mathbf{A}\boldsymbol{\eta} - \boldsymbol{m}^T\mathbf{S}^{-1}\boldsymbol{m}) \tag{6.41}$$

Note that in (6.40), $E(\boldsymbol{\theta})$ is re-written as,

$$\begin{aligned} E(\boldsymbol{\theta}) =& \frac{\beta}{2}\boldsymbol{e}^T\boldsymbol{e} + \frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\eta})^T\mathbf{A}(\boldsymbol{\theta}-\boldsymbol{\eta}) \\ =& \frac{1}{2}\left[\beta\boldsymbol{y}^T\boldsymbol{y} - 2\beta\boldsymbol{\theta}^T\mathbf{P}^T\boldsymbol{y} + \beta\boldsymbol{\theta}^T\mathbf{P}^T\mathbf{P}\boldsymbol{\theta} + \boldsymbol{\theta}^T\mathbf{A}\boldsymbol{\theta} - 2\boldsymbol{\theta}^T\mathbf{A}\boldsymbol{\eta} + \boldsymbol{\eta}^T\mathbf{A}\boldsymbol{\eta}\right] \\ =& \frac{1}{2}\left[\beta\boldsymbol{y}^T\boldsymbol{y} + \boldsymbol{\eta}^T\mathbf{A}\boldsymbol{\eta} + \boldsymbol{\theta}^T(\mathbf{A}+\beta\mathbf{P}^T\mathbf{P})\boldsymbol{\theta} - 2\boldsymbol{\theta}^T(\mathbf{A}\boldsymbol{\eta}+\beta\mathbf{P}^T\boldsymbol{y})\right] \\ =& \frac{1}{2}\left[\beta\boldsymbol{y}^T\boldsymbol{y} + \boldsymbol{\eta}^T\mathbf{A}\boldsymbol{\eta} - \boldsymbol{m}^T\mathbf{S}^{-1}\boldsymbol{m} + (\boldsymbol{\theta}-\boldsymbol{m})^T\mathbf{S}^{-1}(\boldsymbol{\theta}-\boldsymbol{m})\right] \\ =& E(\boldsymbol{y},\boldsymbol{\eta}) + \frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{m})^T\mathbf{S}^{-1}(\boldsymbol{\theta}-\boldsymbol{m}) \end{aligned} \tag{6.42}$$

As can be seen, the second term in Eq (6.40) is the power exponent term of the posterior distribution $\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{m},\mathbf{S})$. The log marginal likelihood is simply written in the form

$$\ln p(\boldsymbol{y}|\mathbf{P},\boldsymbol{\eta},\boldsymbol{\alpha},\beta) = \frac{N}{2}\ln\beta + \frac{1}{2}\sum_{i=1}^{M}\ln\alpha_i - E(\boldsymbol{y},\boldsymbol{\eta}) + \frac{1}{2}\ln|\mathbf{S}| - \frac{N}{2}\ln 2\pi \tag{6.43}$$

To maximise the marginal likelihood function, its derivative with respect to $\boldsymbol{\alpha}$ and β is computed, respectively.

$$\frac{\partial}{\partial\alpha_i}\ln p(\boldsymbol{y}|\mathbf{P}) = \frac{1}{2\alpha_i} - \frac{1}{2}(m_i-\eta_i)^2 - \frac{1}{2}S_{ii} \tag{6.44}$$

$$\frac{\partial}{\partial\beta}\ln p(\boldsymbol{y}|\mathbf{P}) = \frac{1}{2\beta}\left[N - \sum_{i=1}^{M}(1-\alpha_i S_{ii})\right] - \frac{1}{2}\|\boldsymbol{y}-\mathbf{P}\boldsymbol{m}\|_2^2 \tag{6.45}$$

The derivation of (6.44) and (6.45) is given in the following content.

The maximum is obtained when Eqs (6.44) and (6.45) are both equivalent to zero, which gives

$$\alpha_i = \frac{\gamma_i}{(m_i-\eta_i)^2} \tag{6.46}$$

$$\beta^{-1} = \frac{\|\boldsymbol{y}-\mathbf{P}\boldsymbol{m}\|_2^2}{N-\sum_{i=1}^{M}\gamma_i} \tag{6.47}$$

where $\gamma_i = 1-\alpha_i S_{ii}$. Therefore, the regularisation parameter in (6.11) can be given as

$$\mu_i = \frac{\alpha_i}{\beta} = \frac{\gamma_i}{N-\sum_{i=1}^{M}\gamma_i}\frac{\|\boldsymbol{y}-\mathbf{P}\boldsymbol{m}\|_2^2}{(m_i-\eta_i)^2} \tag{6.48}$$

Derivation of (6.44)

The derivative of $E(\boldsymbol{y},\boldsymbol{\eta})$ with respect to α_i is given by

$$\begin{aligned}\frac{\partial}{\partial\alpha_i}E(\boldsymbol{y},\boldsymbol{\eta}) =&\frac{1}{2}\frac{\partial}{\partial\alpha_i}(\boldsymbol{\eta}^T\mathbf{A}\boldsymbol{\eta}) - \frac{1}{2}\frac{\partial}{\partial\alpha_i}(\boldsymbol{m}^T\mathbf{S}^{-1}\boldsymbol{m})\\ =&\frac{1}{2}\eta_i^2 - \frac{1}{2}\frac{\partial}{\partial\alpha_i}(\beta^2\boldsymbol{y}^T\mathbf{PSP}^T\boldsymbol{y}) - \frac{1}{2}\frac{\partial}{\partial\alpha_i}(\boldsymbol{\eta}^T\mathbf{A}^T\mathbf{SA}\boldsymbol{\eta}) - \frac{\partial}{\partial\alpha_i}(\beta\boldsymbol{y}^T\mathbf{PSA}\boldsymbol{\eta})\end{aligned} \tag{6.49}$$

The derivatives of last three terms in (6.49) with respect to α_i are computed respectively as follows,

$$\begin{aligned}\frac{\partial}{\partial\alpha_i}(\beta^2\boldsymbol{y}^T\mathbf{PSP}^T\boldsymbol{y}) =&\mathrm{Tr}\left[\frac{\partial}{\partial\mathbf{S}^{-1}}(\beta^2\boldsymbol{y}^T\mathbf{PSP}^T\boldsymbol{y})\frac{\partial\mathbf{S}^{-1}}{\partial\alpha_i}\right]\\ =&-\mathrm{Tr}\left[\mathbf{S}(\beta\mathbf{P}^T\boldsymbol{y})(\beta\mathbf{P}^T\boldsymbol{y})^T\mathbf{SI}_i\right]\end{aligned} \tag{6.50}$$

$$\begin{aligned}\frac{\partial}{\partial\alpha_i}(\boldsymbol{\eta}^T\mathbf{A}^T\mathbf{SA}\boldsymbol{\eta}) =&2\boldsymbol{\eta}^T\frac{\partial\mathbf{A}^T}{\partial\alpha_i}\mathbf{SA}\boldsymbol{\eta} + \boldsymbol{\eta}^T\mathbf{A}^T\frac{\partial\mathbf{S}}{\partial\alpha_i}\mathbf{A}\boldsymbol{\eta}\\ =&2\eta_i\mathbf{1}_i^T\mathbf{SA}\boldsymbol{\eta} + \boldsymbol{\eta}^T\mathbf{A}^T(-\mathbf{SI}_i\mathbf{S})\mathbf{A}\boldsymbol{\eta}\end{aligned} \tag{6.51}$$

where $\mathbf{1}_i = [0...1...0]^T$ denotes a vector whose ith element is 1 and others are 0.

$$\begin{aligned}\frac{\partial}{\partial\alpha_i}(\beta\boldsymbol{y}^T\mathbf{PSA}\boldsymbol{\eta}) =&\beta\boldsymbol{y}^T\mathbf{P}\frac{\partial\mathbf{S}}{\partial\alpha_i}\mathbf{A}\boldsymbol{\eta} + \beta\boldsymbol{y}^T\mathbf{PS}\frac{\partial\mathbf{A}}{\partial\alpha_i}\boldsymbol{\eta}\\ =&\beta\boldsymbol{y}^T\mathbf{P}(-\mathbf{SI}_i\mathbf{S})\mathbf{A}\boldsymbol{\eta} + \beta\boldsymbol{y}^T\mathbf{PS}\eta_i\mathbf{1}\end{aligned} \tag{6.52}$$

Let $\boldsymbol{\zeta} = \mathbf{SA}\boldsymbol{\eta}$, we can write the derivative of $E(\boldsymbol{y},\boldsymbol{\eta})$ with respect to α_i as,

$$\begin{aligned}\frac{\partial}{\partial\alpha_i}E(\boldsymbol{y},\boldsymbol{\eta}) =&\frac{1}{2}\eta_i^2 + \frac{1}{2}\mathrm{Tr}\left[(\boldsymbol{m}-\boldsymbol{\zeta})(\boldsymbol{m}-\boldsymbol{\zeta})^T\mathbf{I}_i\right] - \boldsymbol{m}^T\eta_i\mathbf{1} + \frac{1}{2}(2\boldsymbol{m}-\boldsymbol{\zeta})^T\mathbf{I}_i\boldsymbol{\zeta}\\ =&\frac{1}{2}\eta_i^2 + \frac{1}{2}(m_i-\zeta_i)^2 - m_i\eta_i + \frac{1}{2}(2m_i-\zeta_i)\zeta_i\\ =&\frac{1}{2}(m_i-\eta_i)^2\end{aligned} \tag{6.53}$$

Given that

$$\frac{\partial}{\partial\alpha_i}\ln|\mathbf{S}| = \mathrm{Tr}\left[\mathbf{S}^{-1}\frac{\partial\mathbf{S}}{\partial\alpha_i}\right] = \mathrm{Tr}\left[-\frac{\partial\mathbf{S}^{-1}}{\partial\alpha_i}\mathbf{S}\right] = -S_{ii} \tag{6.54}$$

the derivative of $\ln p(\boldsymbol{y}|\mathbf{P})$ with respect to α_i is obtained as follows,

$$\frac{\partial}{\partial\alpha_i}\ln p(\boldsymbol{y}|\mathbf{P}) = \frac{1}{2\alpha_i} - \frac{1}{2}(m_i-\eta_i)^2 - \frac{1}{2}S_{ii} \tag{6.55}$$

Derivation of (6.45)

The derivative of $E(\boldsymbol{y},\boldsymbol{\eta})$ with respect to β is given by

$$\begin{aligned}\frac{\partial}{\partial\beta}E(\boldsymbol{y},\boldsymbol{\eta}) =&\frac{1}{2}\boldsymbol{y}^T\boldsymbol{y}-\frac{1}{2}\frac{\partial}{\partial\beta}(\boldsymbol{m}^T\mathbf{S}^{-1}\boldsymbol{m})\\ =&\frac{1}{2}\boldsymbol{y}^T\boldsymbol{y}-\frac{1}{2}\frac{\partial}{\partial\beta}(\beta^2\boldsymbol{y}^T\mathbf{PSP}^T\boldsymbol{y})-\frac{1}{2}\frac{\partial}{\partial\beta}(\boldsymbol{\eta}^T\mathbf{A}^T\mathbf{SA}\boldsymbol{\eta})-\frac{\partial}{\partial\beta}(\beta\boldsymbol{y}^T\mathbf{PSA}\boldsymbol{\eta})\end{aligned} \tag{6.56}$$

The derivatives of last three terms in (6.56) with respect to β are computed respectively as follows,

$$\begin{aligned}\frac{\partial}{\partial\beta}(\beta^2\boldsymbol{y}^T\mathbf{PSP}^T\boldsymbol{y}) =&2\beta\boldsymbol{y}^T\mathbf{PSP}^T\boldsymbol{y}+\beta^2\frac{\partial}{\partial\beta}(\boldsymbol{y}^T\mathbf{PSP}^T\boldsymbol{y})\\ =&2\beta\boldsymbol{y}^T\mathbf{PSP}^T\boldsymbol{y}-\beta^2\boldsymbol{y}^T\mathbf{P}\frac{\partial\mathbf{S}}{\partial\beta}\mathbf{P}^T\boldsymbol{y}\end{aligned} \tag{6.57}$$

$$\frac{\partial}{\partial\beta}(\boldsymbol{\eta}^T\mathbf{A}^T\mathbf{SA}\boldsymbol{\eta}) = \boldsymbol{\eta}^T\mathbf{A}^T\frac{\partial\mathbf{S}}{\partial\beta}\mathbf{A}\boldsymbol{\eta} \tag{6.58}$$

$$\frac{\partial}{\partial\beta}(\beta\boldsymbol{y}^T\mathbf{PSA}\boldsymbol{\eta}) = \boldsymbol{y}^T\mathbf{PSA}\boldsymbol{\eta}+\beta\boldsymbol{y}^T\mathbf{P}\frac{\partial\mathbf{S}}{\partial\beta}\mathbf{A}\boldsymbol{\eta} \tag{6.59}$$

As $\partial\mathbf{S}/\partial\beta = -\mathbf{SP}^T\mathbf{PS}$, we can write the derivative of $E(\boldsymbol{y},\boldsymbol{\eta})$ with respect to β as

$$\frac{\partial}{\partial\beta}E(\boldsymbol{y},\boldsymbol{\eta}) =\frac{1}{2}\boldsymbol{y}^T\boldsymbol{y}-\boldsymbol{y}^T\mathbf{P}\boldsymbol{m}+\frac{1}{2}\boldsymbol{m}^T\mathbf{P}^T\mathbf{P}\boldsymbol{m} = \frac{1}{2}\|\boldsymbol{y}-\mathbf{P}\boldsymbol{m}\|_2^2 \tag{6.60}$$

Given that

$$\frac{\partial}{\partial\beta}\ln|\mathbf{S}| =\mathrm{Tr}\left[\mathbf{S}^{-1}\frac{\partial\mathbf{S}}{\partial\beta}\right] = \mathrm{Tr}\left[-\frac{\partial\mathbf{S}^{-1}}{\partial\beta}\mathbf{S}\right] = -\mathrm{Tr}\left[\mathbf{P}^T\mathbf{PS}\right] = -\beta^{-1}(\mathbf{I}-\mathbf{AS}) \tag{6.61}$$

the derivative of $\ln p(\boldsymbol{y}|\mathbf{P})$ with respect to β is obtained as follows,

$$\frac{\partial}{\partial\beta}\ln p(\boldsymbol{y}|\mathbf{P}) = \frac{1}{2\beta}\left[N-\sum_{i=1}^{M}(1-\alpha_iS_{ii})\right]-\frac{1}{2}\|\boldsymbol{y}-\mathbf{P}\boldsymbol{m}\|_2^2 \tag{6.62}$$

7

Conclusions and future work

7.1 Conclusions

This research project has developed an industrial process monitoring methodology based on a novel time-sensor domain data decomposition and analysis framework. The so-called time and sensor domain components provide a new perspective to analyse the time-varying properties of a production process. Main tasks such as process anomaly detection, time-varying process monitoring, and process states prediction are investigated. This section summarises these findings as follows.

1. A novel time and sensor domain data decomposition and analysis framework is proposed from the time-space decomposition method in DPSs modelling. Under this framework, the multi-sensor signals are decomposed into the TDCs and SDCs such that the original data can be compressed effectively with dominant information remaining. Time series analysis including STL decomposition, AR modelling, and PSD estimation are used to extract condition-sensitive features from TDCs. Then the time and sensor domain features are used to build a fault diagnosis model, whose performance is demonstrated by diagnosing the status of cutting tools in milling experiments. The results verify that the TDC and SDC features can provide at least one of the top three good classification performance in all the cases. Due

to the robustness to the effect of measurement error, the proposed time and sensor decomposition framework shows a better consistency between training and testing data distributions, yielding better generalisation capability compared to many existing feature or decision-layer fusion methods.

2. To overcome the limits of insufficient training data, an unsupervised process anomaly detection approach is proposed based on TSD. The novelty lies in the synthesis of time and sensor domain information. After the data decomposition, representative features are extracted from the TDCs. Meanwhile, the baseline stages and corresponding threshold can be determined and updated adaptively according to the SDC-based clustering analysis. The proposed approach has been applied to the analysis of sensor data from two practical machining experiments for tool breakage detection. Comparative studies with existing methods have also been conducted. The results have demonstrated the effectiveness, the potential in process anomaly detection and the advantages of the new approach over existing techniques in terms of both computation time and overall accuracy in anomaly detection.

3. To address the challenges in time-varying process monitoring, a recursive time-sensor decomposition framework has been developed. Instead of applying TSD to the data blocks separately, RTSD considers the time-dependency among all available data and updates the correlation matrix recursively. As a result, the obtained SDCs can reflect the unknown but deterministic trend of the process variations. Meanwhile, the time domain features represent the overall variations induced by both time-varying trend and faults. A control chart with adaptive control limits is built so as to adapt to the time-varying behaviour of the process. The proposed approach has been validated through a simulation study and an experimental study on the tool wear process monitoring during dynamic milling. The results have demonstrated that the new approach can provide the best trade-off between FDR and FAR. Early false alarms can be avoided, providing a reliable monitoring performance.

4. In data-driven process states predictions, feature selection techniques can avoid overfitting by building a sparse model. To deal with the issue of insufficient sparsity and large computational burden of existing feature selection methods, a novel local regularisation assisted split augmented Lagrangian shrinkage algorithm (LR-SALSA) has been proposed. An individual penalty parameter is introduced to each coefficient such that redundant terms are removed in time. An iterative algorithm, based on the Bayesian evidence framework, is derived to optimise the local regularisation

hyperparameters automatically and guarantee model sparsity. After the important terms are determined, model coefficients are re-estimated by the least squares method. The effectiveness of the proposed approach is demonstrated through both simulation and experimental studies. The results show that the proposed LR-SALSA has a good trade-off among model sparsity, estimation accuracy, and computational efficiency in comparison to five existing feature selection methods. The proposed method can also be generalised to solve the l_1 minimisation problem, leading to a sparse solution to industrial process prediction with high computational efficiency.

These studies demonstrate that many common challenges in process monitoring are addressed by the proposed time-sensor domain data decomposition and analysis framework. All the developed algorithms can be integrated into a PHM system that provides multiple modules from anomaly detection to RUL prediction. The users should select the module according to the objectives and requirements of their tasks. For instance, RTSD is able to detect subtle fault compared to TSD, yet bringing the risk of false alarms. The trade-off between robustness and accuracy can be made through feasibility study. These findings and attempts will extend applications of the proposed methods in various industrial systems including mechanical equipments, chemical processes, metalworking, etc.

7.2 Future work

Although the present study has comprehensively investigated the performance of time-sensor decomposition framework and its application in PHM of cutting tools, there are still many works to develop in the future. These works include theoretical development and practical applications. Details of these possible future research are listed as follows.

1. The proposed novel multi-sensor data decomposition framework can not only reduce the analysis complexity, but also reveal potential time-varying property of industrial processes. This has been validated in the experiments, for example, the sensor domain information is used to determine the baseline passes or update the control limits. In the future, further theoretical analysis can be carried out to provide a deeper understanding of the characteristics of the time and sensor domain components, e.g., quantifying the changes in two domains information, investigating the influence of measurement noises, and so on. The target is to produce more effective ways to synthesise TDCs and SDCs in different scenarios.

2. The industrial processes always show some 'imperfections' violating theoretical assumptions. For instance, the sensor measurements could be nonstationary, time-dependent, and the relation between different signals may be nonlinear. How to improve the time-sensor decomposition framework to handle these complexities is a potential research direction. The Hankel matrix containing delayed time series can reveal dynamical property of systems, and the kernel trick can deal with process nonlinearity by transforming the raw data into a high-dimensional space. These methods might improve the applicability of the time-sensor decomposition algorithm.

3. One of the research objectives is to develop an industry-oriented TCM system. Some practical problems have been considered in this study. However, there is still a gap in the current technology because the real machining process is quite complicated. The variations in machining parameters, part shapes, and tool types would affect sensor signals, making the extracted information show different patterns. In this case, introducing operation-related knowledge to call different models for different processes may improve the adaptability of the monitoring system. Other decisions made from a physical model or vision system can be incorporated to enhance the reliability of monitoring results.

4. Moreover, it is important to highlight that the proposed framework can potentially be

used to analyse multi-sensor signals in industrial processes where a large number of sensors are utilised. This framework offers the advantage of reducing redundant information, thereby better capturing the dominant dynamics of the processes. In addition, the integration of time and sensor domain information improves the understanding of underlying processes, which will guarantee assets safety and optimise production efficiency. The decisions made from the monitoring system can facilitate predictive maintenance of key components.

These future works will extend the understanding and enrich the application scenarios of the proposed methods. In theory, the proposed framework can contribute to the advancement of data-driven process monitoring techniques. In practice, the developed methods will find more opportunities to be implemented in production scenarios. It is expected that more developments are produced under the established framework.

References

[1] Youqing Wang, Yihao Qin, Zhijiang Lou and Xin Ma. 'Survey on Recursive Statistical Process Monitoring Methods'. In: *The Canadian Journal of Chemical Engineering* 100.9 (2022), pp. 2093–2103 (cit. on p. 1).

[2] Tiago Rato, Marco Reis, Eric Schmitt, Mia Hubert and Bart De Ketelaere. 'A Systematic Comparison of PCA-based Statistical Process Monitoring Methods for High-Dimensional, Time-Dependent Processes'. In: *AIChE Journal* 62.5 (2016), pp. 1478–1493 (cit. on pp. 1, 86).

[3] Bart De Ketelaere, Mia Hubert and Eric Schmitt. 'Overview of PCA-Based Statistical Process-Monitoring Methods for Time-Dependent, High-Dimensional Data'. In: *Journal of Quality Technology* 47.4 (Oct. 2015), pp. 318–335 (cit. on pp. 1, 86, 91).

[4] Yihao Qin, Yayun Yan, Hongquan Ji and Youqing Wang. 'Recursive Correlative Statistical Analysis Method With Sliding Windows for Incipient Fault Detection'. In: *IEEE Transactions on Industrial Electronics* 69.4 (Apr. 2022), pp. 4185–4194 (cit. on p. 1).

[5] Chunhui Zhao and Biao Huang. 'Incipient Fault Detection for Complex Industrial Processes with Stationary and Nonstationary Hybrid Characteristics'. In: *Industrial & Engineering Chemistry Research* 57.14 (Apr. 2018), pp. 5045–5057 (cit. on p. 1).

[6] Xuebing Li, Xianli Liu, Caixu Yue, Steven Y. Liang and Lihui Wang. 'Systematic Review on Tool Breakage Monitoring Techniques in Machining Operations'. In: *International Journal of Machine Tools and Manufacture* 176 (May 2022), p. 103882 (cit. on pp. 1, 15, 62).

[7] Nitin Ambhore, Dinesh Kamble, Satish Chinchanikar and Vishal Wayal. 'Tool Condition Monitoring System: A Review'. In: *Materials Today: Proceedings* 2.4-5 (2015), pp. 3419–3428 (cit. on p. 2).

[8] Yuqing Zhou and Wei Xue. 'Review of Tool Condition Monitoring Methods in Milling Processes'. In: *The International Journal of Advanced Manufacturing Technology* 96.5-8 (May 2018), pp. 2509–2523 (cit. on pp. 2, 3, 15, 19, 54, 62, 104).

[9] Y. Altintas and I. Yellowley. 'In-Process Detection of Tool Failure in Milling Using Cutting Force Models'. In: *Journal of Engineering for Industry* 111.2 (May 1989), pp. 149–157 (cit. on pp. 2, 3).

[10] Adam G. Rehorn, Jin Jiang and Peter E. Orban. 'State-of-the-Art Methods and Results in Tool Condition Monitoring: A Review'. In: *The International Journal of Advanced Manufacturing Technology* 26.7-8 (Oct. 2005), pp. 693–710 (cit. on pp. 2, 3, 15).

[11] Jaydeep Karandikar, Tom McLeay, Sam Turner and Tony Schmitz. 'Tool Wear Monitoring Using Naïve Bayes Classifiers'. In: *The International Journal of Advanced Manufacturing Technology* 77.9-12 (Apr. 2015), pp. 1613–1626 (cit. on pp. 2, 33, 34, 37).

[12] H. Shao, H.L. Wang and X.M. Zhao. 'A Cutting Power Model for Tool Wear Monitoring in Milling'. In: *International Journal of Machine Tools and Manufacture* 44.14 (Nov. 2004), pp. 1503–1509 (cit. on pp. 3, 21).

[13] Mehdi Nouri, Barry K. Fussell, Beth L. Ziniti and Ernst Linder. 'Real-Time Tool Wear Monitoring in Milling Using a Cutting Condition Independent Method'. In: *International Journal of Machine Tools and Manufacture* 89 (Feb. 2015), pp. 1–13 (cit. on pp. 3, 21).

[14] Luca Bernini, Paolo Albertelli and Michele Monno. 'Mill Condition Monitoring Based on Instantaneous Identification of Specific Force Coefficients under Variable Cutting Conditions'. In: *Mechanical Systems and Signal Processing* 185 (Feb. 2023), p. 109820 (cit. on pp. 3, 21).

[15] Sultan Binsaeid, Shihab Asfour, Sohyung Cho and Arzu Onar. 'Machine Ensemble Approach for Simultaneous Detection of Transient and Gradual Abnormalities in End Milling Using Multisensor Fusion'. In: *Journal of Materials Processing Technology* 209.10 (June 2009), pp. 4728–4738 (cit. on pp. 4, 27, 33).

[16] Y. Zhou and W. Xue. 'A Multisensor Fusion Method for Tool Condition Monitoring in Milling'. In: *Sensors* 18.11 (2018) (cit. on pp. 4, 27, 35).

[17] Dongdong Kong, Yongjie Chen and Ning Li. 'Gaussian Process Regression for Tool Wear Prediction'. In: *Mechanical Systems and Signal Processing* 104 (May 2018), pp. 556–574 (cit. on pp. 4, 29, 30, 35).

[18] Junhong Zhou. 'Advanced Feature Extraction and Selection for Fault Diagnosis and Prognosis'. PhD thesis. Nanyang Technological University, 2012 (cit. on p. 4).

[19] Zhengyou Xie, Jianguang Li and Yong Lu. 'Feature Selection and a Method to Improve the Performance of Tool Condition Monitoring'. In: *The International Journal of Advanced Manufacturing Technology* 100.9-12 (Feb. 2019), pp. 3197–3206 (cit. on pp. 4, 26).

[20] Weili Cai, Wenjuan Zhang, Xiaofeng Hu and Yingchao Liu. 'A Hybrid Information Model Based on Long Short-Term Memory Network for Tool Condition Monitoring'. In: *Journal of Intelligent Manufacturing* 31.6 (Aug. 2020), pp. 1497–1510 (cit. on pp. 4, 35, 36).

[21] Xiaoyang Zhang, Xin Lu, Weidong Li and Sheng Wang. 'Prediction of the Remaining Useful Life of Cutting Tool Using the Hurst Exponent and CNN-LSTM'. In: *The International Journal of Advanced Manufacturing Technology* 112.7-8 (Feb. 2021), pp. 2277–2299 (cit. on pp. 4, 35).

[22] Kunpeng Zhu, Hao Guo, Si Li and Xin Lin. 'Physics-Informed Deep Learning for Tool Wear Monitoring'. In: *IEEE Transactions on Industrial Informatics* (2023), pp. 1–10 (cit. on p. 4).

[23] Shixu Sun, Xiaofeng Hu and Yingchao Liu. 'An Imbalanced Data Learning Method for Tool Breakage Detection Based on Generative Adversarial Networks'. In: *Journal of Intelligent Manufacturing* 33.8 (Dec. 2022), pp. 2441–2455 (cit. on pp. 4, 62, 63).

[24] F Cannarile, P Baraldi, M Compare et al. 'An Unsupervised Clustering Method for Assessing the Degradation State of Cutting Tools Used in the Packaging Industry'. In: *Safety and Reliability – Theory and Applications*. High Tatras Mountains, Tatranské Matliare, Slovak Republic: CRC Press, June 2017, pp. 138–138 (cit. on pp. 4, 32).

[25] Thomas McLeay, Michael S Turner and Keith Worden. 'A Novel Approach to Machining Process Fault Detection Using Unsupervised Learning'. In: *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* (Sept. 2020), p. 095440542093755 (cit. on pp. 4, 27, 62, 66).

[26] Genyi Luo, J.E.D. Hurwitz and Thomas G. Habetler. 'A Survey of Multi-Sensor Systems for Online Fault Detection of Electric Machines'. In: *2019 IEEE 12th International Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives (SDEMPED)*. Aug. 2019, pp. 338–343 (cit. on pp. 4, 49).

[27] Han-Xiong Li, Chenkun Qi and Yongguang Yu. 'A Spatio-Temporal Volterra Modeling Approach for a Class of Distributed Industrial Processes'. In: *Journal of Process Control* 19.7 (July 2009), pp. 1126–1142 (cit. on pp. 4, 39, 40).

[28] Han-Xiong Li and Chenkun Qi. *Spatio-Temporal Modeling of Nonlinear Distributed Parameter Systems: A Time/Space Separation Based Approach*. Intelligent Systems, Control and Automation: Science and Engineering 50. Dordrecht: Springer Science+ Business Media B.V, 2011 (cit. on pp. 4, 5, 39).

[29] Andrew J. Newman. *Model Reduction via the Karhunen-Loeve Expansion Part I: An Exposition*. Tech. rep. 1996 (cit. on pp. 5, 45).

[30] C. R. Featherston, M. J. Gregory, A. Gill and E. O'Sullivan. 'High Value Manufacturing in the UK: A Study of Its Challenges, Opportunities and Emerging Technologies'. In: *Applied Science and Engineering Progress* 6.2 (Dec. 2013), pp. 61–70 (cit. on p. 13).

[31] Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld and Michael Hoffmann. 'Industry 4.0'. In: *Business & Information Systems Engineering* 6.4 (Aug. 2014), pp. 239–242 (cit. on p. 13).

[32] Kunpeng Zhu, Guochao Li and Yu Zhang. 'Big Data Oriented Smart Tool Condition Monitoring System'. In: *IEEE Transactions on Industrial Informatics* 16.6 (June 2020), pp. 4007–4016 (cit. on pp. 14, 19, 112).

[33] *Tool and Process Monitoring*. https://new.artis.de/eng/application/monitoring-solutions-for-machining-center (cit. on p. 14).

[34] *Digital Machining with CoroPlus®*. https://www.sandvik.coromant.com/en-gb/campaigns/coroplus (cit. on p. 14).

[35] *TMAC - Adaptive Tool Monitoring System for CNC Machining*. https://www.caroneng.com/products/tmac/ (cit. on p. 15).

[36] *Nordmann Tool Monitoring*. https://www.toolmonitoring.com/index.html (cit. on p. 15).

[37] *Tool- and Process Monitoring*. https://www.montronix.com/en/products/applications/tool-and-process-monitoring.html (cit. on p. 15).

[38] *Full Process Control with ToolScope*. https://cuttingtools.ceratizit.com/gb/en/tool-solutions/toolscope.html (cit. on p. 15).

[39] Kunpeng Zhu, Yoke San Wong and Geok Soon Hong. 'Wavelet Analysis of Sensor Signals for Tool Condition Monitoring: A Review and Some New Results'. In: *International Journal of Machine Tools and Manufacture* 49.7-8 (June 2009), pp. 537–553 (cit. on pp. 15, 23).

[40] G. Serin, B. Sener, A. M. Ozbayoglu and H. O. Unver. 'Review of Tool Condition Monitoring in Machining and Opportunities for Deep Learning'. In: *The International Journal of Advanced Manufacturing Technology* 109.3-4 (July 2020), pp. 953–974 (cit. on pp. 15, 33, 34, 36).

[41] Wenchao Xiao, Jianghua Huang, Baoyu Wang and Hongchao Ji. 'A Systematic Review of Artificial Intelligence in the Detection of Cutting Tool Breakage in Machining Operations'. In: *Measurement* 190 (Feb. 2022), p. 110748 (cit. on p. 15).

[42] P.W Prickett and C Johns. 'An Overview of Approaches to End Milling Tool Monitoring'. In: *International Journal of Machine Tools and Manufacture* 39.1 (Jan. 1999), pp. 105–122 (cit. on pp. 16–18, 22).

[43] Mei Wang and Jie Wang. 'CHMM for Tool Condition Monitoring and Remaining Useful Life Prediction'. In: *The International Journal of Advanced Manufacturing Technology* 59.5-8 (Mar. 2012), pp. 463–471 (cit. on p. 16).

[44] S.N. Huang, K.K. Tan, Y.S. Wong et al. 'Tool Wear Detection and Fault Diagnosis Based on Cutting Force Monitoring'. In: *International Journal of Machine Tools and Manufacture* 47.3-4 (Mar. 2007), pp. 444–451 (cit. on p. 16).

[45] Dong Gao, Zhirong Liao, Zekun Lv and Yong Lu. 'Multi-Scale Statistical Signal Processing of Cutting Force in Cutting Tool Condition Monitoring'. In: *The International Journal of Advanced Manufacturing Technology* 80.9-12 (Oct. 2015), pp. 1843–1853 (cit. on p. 16).

[46] S.V. Kamarthi and S. Pittner. 'FOURIER AND WAVELET TRANSFORM FOR FLANK WEAR ESTIMATION — A COMPARISON'. In: *Mechanical Systems and Signal Processing* 11.6 (Nov. 1997), pp. 791–809 (cit. on p. 17).

[47] I. Yesilyurt and H. Ozturk. 'Tool Condition Monitoring in Milling Using Vibration Analysis'. In: *International Journal of Production Research* 45.4 (Feb. 2007), pp. 1013–1028 (cit. on p. 17).

[48] Gu-Seon Kang, Seung-Gi Kim, Gi-Dong Yang, Kyung-Hee Park and Dong Yoon Lee. 'Tool Chipping Detection Using Peak Period of Spindle Vibration During End-Milling of Inconel 718'. In: *International Journal of Precision Engineering and Manufacturing* 20.11 (Nov. 2019), pp. 1851–1859 (cit. on pp. 17, 62).

[49] Zhengyou Xie, Jianguang Li and Yong Lu. 'An Integrated Wireless Vibration Sensing Tool Holder for Milling Tool Condition Monitoring'. In: *The International Journal of Advanced Manufacturing Technology* 95.5-8 (Mar. 2018), pp. 2885–2896 (cit. on p. 17).

[50] R. Teti, K. Jemielniak, G. O'Donnell and D. Dornfeld. 'Advanced Monitoring of Machining Operations'. In: *CIRP Annals* 59.2 (2010), pp. 717–739 (cit. on p. 17).

[51] Othman Belgassim and Krzysztof Jemielniak. ‘Tool Failure Detection Based on Statistical Analysis of Metal Cutting Acoustic Emission Signals’. In: *World Academy of Science, Engineering and Technology* 74 (2011), pp. 551–558 (cit. on p. 18).

[52] M.S.H. Bhuiyan, I.A. Choudhury, M. Dahari, Y. Nukman and S.Z. Dawal. ‘Application of Acoustic Emission Sensor to Investigate the Frequency of Tool Wear and Plastic Deformation in Tool Condition Monitoring’. In: *Measurement* 92 (Oct. 2016), pp. 208–217 (cit. on p. 18).

[53] Thomas E McLeay. ‘Unsupervised Monitoring of Machining Processes’. PhD thesis. The University of Sheffield, 2016 (cit. on pp. 18, 31, 32, 37).

[54] X. Li, A. Djordjevich and P.K. Venuvinod. ‘Current-Sensor-Based Feed Cutting Force Intelligent Estimation and Tool Wear Condition Monitoring’. In: *IEEE Transactions on Industrial Electronics* 47.3 (June 2000), pp. 697–702 (cit. on p. 18).

[55] Cyril Drouillet, Jaydeep Karandikar, Chandra Nath et al. ‘Tool Life Predictions in Milling Using Spindle Power with the Neural Network Technique’. In: *Journal of Manufacturing Processes* 22 (Apr. 2016), pp. 161–168 (cit. on pp. 18, 34, 35).

[56] Bin Shen, Yufei Gui, Biao Chen et al. ‘Application of Spindle Power Signals in Tool Condition Monitoring Based on HHT Algorithm’. In: *The International Journal of Advanced Manufacturing Technology* 106.3-4 (Jan. 2020), pp. 1385–1395 (cit. on pp. 18, 25, 62).

[57] B. Y. Lee and Y. S. Not Available. ‘Application of the Discrete Wavelet Transform to the Monitoring of Tool Failure in End Milling Using the Spindle Motor Current’. In: *The International Journal of Advanced Manufacturing Technology* 15.4 (Apr. 1999), pp. 238–243 (cit. on p. 19).

[58] Meng-Kun Liu, Yi-Heng Tseng and Minh-Quang Tran. ‘Tool Wear Monitoring and Prediction Based on Sound Signal’. In: *The International Journal of Advanced Manufacturing Technology* 103.9-12 (Aug. 2019), pp. 3361–3373 (cit. on p. 19).

[59] S.K Choudhury and G Bartarya. ‘Role of Temperature and Surface Finish in Predicting Tool Wear Using Neural Network and Design of Experiments’. In: *International Journal of Machine Tools and Manufacture* 43.7 (May 2003), pp. 747–753 (cit. on p. 19).

[60] F.N. Koumboulis, G.P. Petropoulos and C.S. Mavridis. ‘Fault Detection in Machining via Torque Estimation’. In: *IFAC Proceedings Volumes* 33.20 (July 2000), pp. 217–221 (cit. on p. 19).

[61] S. H. Yeo. ‘Tool Condition Monitoring Using Reflectance of Chip Surface and Neural Network’. In: *Journal of Intelligent Manufacturing* 11.6 (2000), pp. 507–514 (cit. on p. 19).

[62] Sohyung Cho, Sultan Binsaeid and Shihab Asfour. 'Design of Multisensor Fusion-Based Tool Condition Monitoring System in End Milling'. In: *The International Journal of Advanced Manufacturing Technology* 46.5-8 (Jan. 2010), pp. 681–694 (cit. on pp. 19, 26, 33, 37).

[63] Cuneyt Aliustaoglu, H. Metin Ertunc and Hasan Ocak. 'Tool Wear Condition Monitoring Using a Sensor Fusion Model Based on Fuzzy Inference System'. In: *Mechanical Systems and Signal Processing* 23.2 (Feb. 2009), pp. 539–546 (cit. on p. 19).

[64] N. Ghosh, Y.B. Ravi, A. Patra et al. 'Estimation of Tool Wear during CNC Milling Using Neural Network-Based Sensor Fusion'. In: *Mechanical Systems and Signal Processing* 21.1 (Jan. 2007), pp. 466–479 (cit. on pp. 19, 29, 30).

[65] Onur Surucu, Stephen Andrew Gadsden and John Yawney. 'Condition Monitoring Using Machine Learning: A Review of Theory, Applications, and Recent Advances'. In: *Expert Systems with Applications* 221 (July 2023), p. 119738 (cit. on p. 21).

[66] S. Y. Liang and D. A. Dornfeld. 'Tool Wear Detection Using Time Series Analysis of Acoustic Emission'. In: *Journal of Engineering for Industry* 111.3 (Aug. 1989), pp. 199–205 (cit. on p. 21).

[67] Yusuf Altintas. 'In-Process Detection of Tool Breakages Using Time Series Monitoring of Cutting Forces'. In: *International Journal of Machine Tools and Manufacture* 28.2 (Jan. 1988), pp. 157–172 (cit. on p. 22).

[68] Zepeng Liu, Zi-Qiang Lang, Yun-Peng Zhu et al. 'Sensor Data Modeling and Model Frequency Analysis for Detecting Cutting Tool Anomalies in Machining'. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2022), pp. 1–13 (cit. on p. 22).

[69] Bovic Kilundu, Pierre Dehombreux and Xavier Chiementin. 'Tool Wear Monitoring by Machine Learning Techniques and Singular Spectrum Analysis'. In: *Mechanical Systems and Signal Processing* 25.1 (Jan. 2011), pp. 400–415 (cit. on p. 23).

[70] C. S. Ai, Y. J. Sun, G. W. He et al. 'The Milling Tool Wear Monitoring Using the Acoustic Spectrum'. In: *The International Journal of Advanced Manufacturing Technology* 61.5-8 (July 2012), pp. 457–463 (cit. on p. 23).

[71] S.G. Mallat. 'A Theory for Multiresolution Signal Decomposition: The Wavelet Representation'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11.7 (July 1989), pp. 674–693 (cit. on p. 24).

[72] Amin Jahromi Torabi, Meng Joo Er, Xiang Li, Beng Siong Lim and Gan Oon Peen. 'Application of Clustering Methods for Online Tool Condition Monitoring and Fault Diagnosis in High-Speed Milling Processes'. In: *IEEE Systems Journal* 10.2 (June 2016), pp. 721–732 (cit. on p. 24).

[73] C. K. Madhusudana, Hemantha Kumar and S. Narendranath. ‘Face Milling Tool Condition Monitoring Using Sound Signal’. In: *International Journal of System Assurance Engineering and Management* 8.S2 (Nov. 2017), pp. 1643–1653 (cit. on p. 24).

[74] Hamed Rafezi, Javad Akbari and Mehdi Behzad. ‘Tool Condition Monitoring Based on Sound and Vibration Analysis and Wavelet Packet Decomposition’. In: *2012 8th International Symposium on Mechatronics and Its Applications*. Sharjah, United Arab Emirates: IEEE, Apr. 2012, pp. 1–4 (cit. on p. 24).

[75] Norden E Huang, Samuel S Shen and World Scientific (Firm). *Hilbert-Huang Transform and Its Applications*. Singapore; Hakensack, N.J.: World Scientific Pub. Co., 2014 (cit. on p. 24).

[76] Xinhua Shi, Ran Wang, Quntao Chen and Hua Shao. ‘Cutting Sound Signal Processing for Tool Breakage Detection in Face Milling Based on Empirical Mode Decomposition and Independent Component Analysis’. In: *Journal of Vibration and Control* 21.16 (Dec. 2015), pp. 3348–3358 (cit. on p. 25).

[77] Girish Chandrashekar and Ferat Sahin. ‘A Survey on Feature Selection Methods’. In: *Computers & Electrical Engineering* 40.1 (Jan. 2014), pp. 16–28 (cit. on p. 26).

[78] M. Dash and H. Liu. ‘Feature Selection for Classification’. In: *Intelligent data analysis* (1997) (cit. on p. 26).

[79] Rui Zhang, Feiping Nie, Xuelong Li and Xian Wei. ‘Feature Selection with Multi-View Data: A Survey’. In: *Information Fusion* 50 (Oct. 2019), pp. 158–167 (cit. on pp. 26, 28).

[80] C. Scheffer and P.S. Heyns. ‘An Industrial Tool Wear Monitoring System for Interrupted Turning’. In: *Mechanical Systems and Signal Processing* 18.5 (Sept. 2004), pp. 1219–1242 (cit. on p. 26).

[81] Zheng Zhao, Lei Wang, Huan Liu and Jieping Ye. ‘On Similarity Preserving Feature Selection’. In: *IEEE Transactions on Knowledge and Data Engineering* 25.3 (Mar. 2013), pp. 619–632 (cit. on p. 26).

[82] Matthieu Lucke, Xueyu Mei, Anna Stief, Moncef Chioua and Nina F. Thornhill. ‘Variable Selection for Fault Detection and Identification Based on Mutual Information of Alarm Series’. In: *IFAC-PapersOnLine* 52.1 (2019), pp. 673–678 (cit. on p. 26).

[83] Shuaijie Mei, Mei Yuan, Jin Cui, Shaopeng Dong and Juanru Zhao. ‘Machinery Condition Monitoring in the Era of Industry 4.0: A Relative Degree of Contribution Feature Selection and Deep Residual Network Combined Approach’. In: *Computers & Industrial Engineering* 168 (June 2022), p. 108129 (cit. on p. 26).

[84] K. P. Zhu, G. S. Hong and Y. S. Wong. 'A Comparative Study of Feature Selection for Hidden Markov Model-Based Micro-Milling Tool Wear Monitoring'. In: *Machining Science and Technology* 12.3 (Sept. 2008), pp. 348–369 (cit. on p. 26).

[85] Jun-Hong Zhou, Chee Khiang Pang, F.L. Lewis and Zhao-Wei Zhong. 'Intelligent Diagnosis and Prognosis of Tool Wear Using Dominant Feature Identification'. In: *IEEE Transactions on Industrial Informatics* 5.4 (Nov. 2009), pp. 454–464 (cit. on pp. 27, 49, 55).

[86] Jun-Hong Zhou, Chee Khiang Pang, Zhao-Wei Zhong and Frank L. Lewis. 'Tool Wear Monitoring Using Acoustic Emissions by Dominant-Feature Identification'. In: *IEEE Transactions on Instrumentation and Measurement* 60.2 (Feb. 2011), pp. 547–559 (cit. on pp. 27, 62).

[87] A. Malhi and R. X. Gao. 'PCA-based Feature Selection Scheme for Machine Defect Classification'. In: *IEEE Transactions on Instrumentation and Measurement* 53.6 (Dec. 2004), pp. 1517–1525 (cit. on pp. 27, 37, 49, 55, 62).

[88] Meng Hu, Weiwei Ming, Qinglong An and Ming Chen. 'Tool Wear Monitoring in Milling of Titanium Alloy Ti–6Al–4 V under MQL Conditions Based on a New Tool Wear Categorization Method'. In: *The International Journal of Advanced Manufacturing Technology* 104.9-12 (Oct. 2019), pp. 4117–4128 (cit. on p. 27).

[89] Zexuan Zhu, Yew-Soon Ong and Manoranjan Dash. 'Wrapper–Filter Feature Selection Algorithm Using a Memetic Framework'. In: *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)* 37.1 (Feb. 2007), pp. 70–76 (cit. on p. 27).

[90] T. Warren Liao. 'Feature Extraction and Selection from Acoustic Emission Signals with an Application in Grinding Wheel Condition Monitoring'. In: *Engineering Applications of Artificial Intelligence* 23.1 (Feb. 2010), pp. 74–84 (cit. on p. 27).

[91] Xiaoping Liao, Gang Zhou, Zhenkun Zhang, Juan Lu and Junyan Ma. 'Tool Wear State Recognition Based on GWO–SVM with Feature Selection of Genetic Algorithm'. In: *The International Journal of Advanced Manufacturing Technology* 104.1-4 (Sept. 2019), pp. 1051–1063 (cit. on p. 27).

[92] Jundong Li, Kewei Cheng, Suhang Wang et al. 'Feature Selection: A Data Perspective'. In: *ACM Computing Surveys* 50.6 (Nov. 2018), pp. 1–45 (cit. on pp. 28, 112).

[93] Jianbo Yu. 'Tool Condition Prognostics Using Logistic Regression with Penalization and Manifold Regularization'. In: *Applied Soft Computing* 64 (Mar. 2018), pp. 454–467 (cit. on p. 28).

[94] Qibo Yang, Jaskaran Singh and Jay Lee. 'Isolation-Based Feature Selection for Unsupervised Outlier Detection'. In: *Annual Conference of the PHM Society* 11.1 (Sept. 2019) (cit. on p. 28).

[95] Wange Li, Long Chen, Jun Zhao and Wei Wang. 'Embedded Feature Selection Based on Relevance Vector Machines With an Approximated Marginal Likelihood and Its Industrial Application'. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2021), pp. 1–14 (cit. on pp. 28, 113).

[96] Alessandra Caggiano. 'Tool Wear Prediction in Ti-6Al-4V Machining through Multiple Sensor Monitoring and PCA Features Pattern Recognition'. In: *Sensors* 18.3 (Mar. 2018), p. 823 (cit. on pp. 29, 49, 55).

[97] N. Wu and J. Zhang. 'Factor Analysis Based Anomaly Detection'. In: *IEEE Systems, Man and Cybernetics SocietyInformation Assurance Workshop, 2003.* June 2003, pp. 108–115 (cit. on p. 29).

[98] Jinjiang Wang, Junyao Xie, Rui Zhao, Kezhi Mao and Laibin Zhang. 'A New Probabilistic Kernel Factor Analysis for Multisensory Data Fusion: Application to Tool Condition Monitoring'. In: *IEEE Transactions on Instrumentation and Measurement* 65.11 (Nov. 2016), pp. 2527–2537 (cit. on pp. 29, 35, 62).

[99] Shang-Liang Chen and Y.W. Jen. 'Data Fusion Neural Network for Tool Condition Monitoring in CNC Milling Machining'. In: *International Journal of Machine Tools and Manufacture* 40.3 (Feb. 2000), pp. 381–400 (cit. on p. 29).

[100] Víctor H. Jaramillo, James R. Ottewill, Rafał Dudek, Dariusz Lepiarczyk and Paweł Pawlik. 'Condition Monitoring of Distributed Systems Using Two-Stage Bayesian Inference Data Fusion'. In: *Mechanical Systems and Signal Processing* 87 (Mar. 2017), pp. 91–110 (cit. on pp. 29, 30).

[101] Laura I. Burke. 'AN UNSUPERVISED NEURAL NETWORK APPROACH TO TOOL WEAR IDENTIFICATION'. In: *IIE Transactions* 25.1 (Jan. 1993), pp. 16–25 (cit. on p. 31).

[102] V.B. Jammu, K. Danai and S. Malkin. 'Unsupervised Neural Network for Tool Breakage Detection in Turning'. In: *CIRP Annals - Manufacturing Technology* 42.1 (Jan. 1993), pp. 67–70 (cit. on p. 31).

[103] S. Li and M.A. Elbestawi. 'FUZZY CLUSTERING FOR AUTOMATED TOOL CONDITION MONITORING IN MACHINING'. In: *Mechanical Systems and Signal Processing* 10.5 (Sept. 1996), pp. 533–550 (cit. on p. 32).

[104] Jianming Dou, Shengjie Jiao, Chuangwen Xu et al. 'Unsupervised Online Prediction of Tool Wear Values Using Force Model Coefficients in Milling'. In: *The International Journal of Advanced Manufacturing Technology* 109.3-4 (July 2020), pp. 1153–1166 (cit. on p. 32).

[105] M. Grasso, B.M. Colosimo and M. Pacella. 'Profile Monitoring via Sensor Fusion: The Use of PCA Methods for Multi-Channel Data'. In: *International Journal of Production Research* 52.20 (Oct. 2014), pp. 6110–6135 (cit. on pp. 32, 49, 90, 99, 101–103).

[106] Teng Wang, Guoliang Lu and Peng Yan. 'Multi-Sensors Based Condition Monitoring of Rotary Machines: An Approach of Multidimensional Time-Series Analysis'. In: *Measurement* 134 (Feb. 2019), pp. 326–335 (cit. on p. 32).

[107] Tony Boutros and Ming Liang. 'Detection and Diagnosis of Bearing and Cutting Tool Faults Using Hidden Markov Models'. In: *Mechanical Systems and Signal Processing* 25.6 (Aug. 2011), pp. 2102–2124 (cit. on p. 33).

[108] Wan-Hao Hsieh, Ming-Chyuan Lu and Shean-Juinn Chiou. 'Application of Backpropagation Neural Network for Spindle Vibration-Based Tool Wear Monitoring in Micro-Milling'. In: *The International Journal of Advanced Manufacturing Technology* 61.1-4 (July 2012), pp. 53–61 (cit. on p. 33).

[109] Han Ding, Robert X. Gao, Alf J. Isaksson et al. 'State of AI-Based Monitoring in Smart Manufacturing and Introduction to Focused Section'. In: *IEEE/ASME Transactions on Mechatronics* 25.5 (Oct. 2020), pp. 2143–2154 (cit. on p. 33).

[110] Rui Zhao, Ruqiang Yan, Zhenghua Chen et al. 'Deep Learning and Its Applications to Machine Health Monitoring'. In: *Mechanical Systems and Signal Processing* 115 (Jan. 2019), pp. 213–237 (cit. on p. 33).

[111] Chengming Shi, George Panoutsos, Bo Luo et al. 'Using Multiple-Feature-Spaces-Based Deep Learning for Tool Condition Monitoring in Ultraprecision Manufacturing'. In: *IEEE Transactions on Industrial Electronics* 66.5 (May 2019), pp. 3794–3803 (cit. on pp. 33, 37).

[112] Dongfeng Shi and Nabil N. Gindy. 'Tool Wear Predictive Model Based on Least Squares Support Vector Machines'. In: *Mechanical Systems and Signal Processing* 21.4 (May 2007), pp. 1799–1814 (cit. on p. 34).

[113] Yingguang Li, Changqing Liu, Jiaqi Hua, James Gao and Paul Maropoulos. 'A Novel Method for Accurately Monitoring and Predicting Tool Wear under Varying Cutting Conditions Based on Meta-Learning'. In: *CIRP Annals* 68.1 (2019), pp. 487–490 (cit. on pp. 34, 35).

[114] Chuang Sun, Meng Ma, Zhibin Zhao et al. 'Deep Transfer Learning Based on Sparse Autoencoder for Remaining Useful Life Prediction of Tool in Manufacturing'. In: *IEEE Transactions on Industrial Informatics* 15.4 (Apr. 2019), pp. 2416–2425 (cit. on pp. 35, 36).

[115] X Li, B S Lim, J H Zhou et al. 'Fuzzy Neural Network Modelling for Tool Wear Estimation in Dry Milling Operation'. In: *Proceedings of the Annual Conference of the PHM Society 2009*. Vol. 1. 1. 2009, pp. 1–11 (cit. on pp. 35, 131, 132).

[116] Tarak Benkedjouh, N. Zerhouni and S. Rechak. 'Tool Wear Condition Monitoring Based on Continuous Wavelet Transform and Blind Source Separation'. In: *The International Journal of Advanced Manufacturing Technology* 97.9-12 (Aug. 2018), pp. 3311–3323 (cit. on pp. 35, 37).

[117] Xiao Qin, Weizhi Huang, Xuefei Wang, Zezhi Tang and Zepeng Liu. 'Real-Time Remaining Useful Life Prediction of Cutting Tools Using Sparse Augmented Lagrangian Analysis and Gaussian Process Regression'. In: *Sensors* 23.1 (Dec. 2022), p. 413 (cit. on pp. 35, 113).

[118] P. J. Courtois. 'On Time and Space Decomposition of Complex Structures'. In: *Communications of the ACM* 28.6 (June 1985), pp. 590–603 (cit. on p. 40).

[119] Carl Eckart and Gale Young. 'A Principal Axis Transformation for Non-Hermitian Matrices'. In: *Bulletin of the American Mathematical Society* 45.2 (Feb. 1939), pp. 118–122 (cit. on p. 40).

[120] Karl Pearson. 'On Lines and Planes of Closest Fit to Systems of Points in Space'. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (Nov. 1901), pp. 559–572 (cit. on p. 40).

[121] H. Hotelling. 'Analysis of a Complex of Statistical Variables into Principal Components.' In: *Journal of Educational Psychology* 24.6 (1933), pp. 417–441 (cit. on p. 40).

[122] Harold Hotelling. 'Simplified Calculation of Principal Components'. In: *Psychometrika* 1.1 (Mar. 1936), pp. 27–35 (cit. on p. 40).

[123] Kuan Lu, Yulin Jin, Yushu Chen et al. 'Review for Order Reduction Based on Proper Orthogonal Decomposition and Outlooks of Applications in Mechanical Systems'. In: *Mechanical Systems and Signal Processing* 123 (May 2019), pp. 264–297 (cit. on p. 40).

[124] D. D. Kosambi. 'Statistics in Function Space'. In: *D.D. Kosambi: Selected Works in Mathematics and Statistics*. Ed. by Ramakrishna Ramaswamy. New Delhi: Springer India, 2016, pp. 115–123 (cit. on p. 40).

[125] M Loève. *Probability Theory : Foundations, Random Sequences*. New York : Van Nostrand, 1955 (cit. on p. 40).

[126] C. M. Cheng, X. J. Dong, Z. K. Peng, W. M. Zhang and G. Meng. 'Wavelet Basis Expansion-Based Spatio-Temporal Volterra Kernels Identification for Nonlinear Distributed Parameter Systems'. In: *Nonlinear Dynamics* 78.2 (Oct. 2014), pp. 1179–1192 (cit. on p. 40).

[127] J. L Lumley, P. Holmes, G. Berkooz and C. W. Rowley. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. United Kingdom: Cambridge University Press., 2012 (cit. on p. 40).

[128] Bing-Chuan Wang and Han-Xiong Li. 'A Sliding Window Based Dynamic Spatiotemporal Modeling for Distributed Parameter Systems With Time-Dependent Boundary Conditions'. In: *IEEE Transactions on Industrial Informatics* 15.4 (Apr. 2019), pp. 2044–2053 (cit. on p. 40).

[129] Ning Li, Chen Hua, Haifeng Wang, Shaoyuan Li and Shuzhi Sam Ge. 'Time–Space Decomposition-Based Generalized Predictive Control of a Transport-Reaction Process'. In: *Industrial & Engineering Chemistry Research* 50.20 (Oct. 2011), pp. 11628–11635 (cit. on p. 40).

[130] B.F. Feeny and R. Kappagantu. 'ON THE PHYSICAL INTERPRETATION OF PROPER ORTHOGONAL MODES IN VIBRATIONS'. In: *Journal of Sound and Vibration* 211.4 (Apr. 1998), pp. 607–616 (cit. on p. 41).

[131] M.F.A. Azeez and A.F. Vakakis. 'PROPER ORTHOGONAL DECOMPOSITION (POD) OF A CLASS OF VIBROIMPACT OSCILLATIONS'. In: *Journal of Sound and Vibration* 240.5 (Mar. 2001), pp. 859–889 (cit. on p. 41).

[132] Y.C. Liang, H.P. Lee, S.P. Lim et al. 'PROPER ORTHOGONAL DECOMPOSITION AND ITS APPLICATIONS—PART I: THEORY'. In: *Journal of Sound and Vibration* 252.3 (May 2002), pp. 527–544 (cit. on p. 41).

[133] Paolo Tiso and Daniel J. Rixen. 'Reduction Methods for MEMS Nonlinear Dynamic Analysis'. In: *Nonlinear Modeling and Applications, Volume 2*. Ed. by Tom Proulx. New York, NY: Springer New York, 2011, pp. 53–65 (cit. on p. 41).

[134] B.W. Ang, Bin Su and H. Wang. 'A Spatial–Temporal Decomposition Approach to Performance Assessment in Energy and Emissions'. In: *Energy Economics* 60 (Nov. 2016), pp. 112–121 (cit. on p. 41).

[135] Chen Chen, Tao Zhao, Rong Yuan and Yingchao Kong. 'A Spatial-Temporal Decomposition Analysis of China's Carbon Intensity from the Economic Perspective'. In: *Journal of Cleaner Production* 215 (Apr. 2019), pp. 557–569 (cit. on p. 41).

[136] Diego Bueso, Maria Piles and Gustau Camps-Valls. ‘Nonlinear PCA for Spatio-Temporal Analysis of Earth Observation Data’. In: *IEEE Transactions on Geoscience and Remote Sensing* 58.8 (Aug. 2020), pp. 5752–5763 (cit. on p. 41).

[137] Francisco Castells, Pablo Laguna, Leif Sörnmo, Andreas Bollmann and José Millet Roig. ‘Principal Component Analysis in ECG Signal Processing’. In: *EURASIP Journal on Advances in Signal Processing* 2007.1 (Dec. 2007), p. 074580 (cit. on p. 41).

[138] Dan Foti, Greg Hajcak and Joseph Dien. ‘Differentiating Neural Responses to Emotional Pictures: Evidence from Temporal-Spatial PCA’. In: *Psychophysiology* 46.3 (May 2009), pp. 521–530 (cit. on p. 41).

[139] Irem Y. Tumer, Kristin L. Wood and Ilene J. Busch-Vishniac. ‘Improving Manufacturing Precision Using the Karhunen-Loève Transform’. In: *Volume 4: Design for Manufacturing Conference*. Sacramento, California, USA: American Society of Mechanical Engineers, Sept. 1997, V004T31A046 (cit. on p. 41).

[140] Irem Y. Tumer, Kristin L. Wood and Ilene J. Busch-Vishniac. ‘Condition Monitoring Methodology for Manufacturing and Design’. In: *Volume 4: 3rd Design for Manufacturing Conference*. Atlanta, Georgia, USA: American Society of Mechanical Engineers, Sept. 1998, V004T04A034 (cit. on pp. 41, 42).

[141] Irem Y. Tumer, Kristin L. Wood and Ilene J. Busch-Vishniac. ‘A Mathematical Transform to Analyze Part Surface Quality in Manufacturing’. In: *Journal of Manufacturing Science and Engineering* 122.1 (Feb. 2000), pp. 273–279 (cit. on p. 41).

[142] Irem Y. Tumer and Edward M. Huff. ‘Using Triaxial Accelerometer Data for Vibration Monitoring of Helicopter Gearboxes’. In: *Volume 6C: 18th Biennial Conference on Mechanical Vibration and Noise*. Pittsburgh, Pennsylvania, USA: American Society of Mechanical Engineers, Sept. 2001, pp. 3161–3171 (cit. on pp. 41, 43).

[143] Irem Y. Tumer and Edward M. Huff. ‘Analysis of Triaxial Vibration Data for Health Monitoring of Helicopter Gearboxes’. In: *Journal of Vibration and Acoustics* 125.1 (Jan. 2003), pp. 120–128 (cit. on p. 41).

[144] RB Cleveland. ‘STL: A Seasonal-Trend Decomposition Procedure Based on Loess’. In: *Journal of Official Statistics* (1990) (cit. on p. 47).

[145] Paul Shaman and Robert A. Stine. ‘The Bias of Autoregressive Coefficient Estimators’. In: *Journal of the American Statistical Association* 83.403 (Sept. 1988), pp. 842–848 (cit. on p. 47).

[146] K. Verbert, R. Babuška and B. De Schutter. ‘Bayesian and Dempster–Shafer Reasoning for Knowledge-Based Fault Diagnosis–A Comparative Study’. In: *Engineering Applications of Artificial Intelligence* 60 (Apr. 2017), pp. 136–150 (cit. on p. 49).

[147] Cheng-Hsuan Li, Chin-Teng Lin, Bor-Chen Kuo and Hui-Shan Chu. 'An Automatic Method for Selecting the Parameter of the RBF Kernel Function to Support Vector Machines'. In: *2010 IEEE International Geoscience and Remote Sensing Symposium*. Honolulu, HI, USA: IEEE, July 2010, pp. 836–839 (cit. on p. 53).

[148] Enrique F. Schisterman, Neil J. Perkins, Aiyi Liu and Howard Bondell. 'Optimal Cut-Point and Its Corresponding Youden Index to Discriminate Individuals Using Pooled Blood Samples'. In: *Epidemiology* 16.1 (2005), pp. 73–81 (cit. on p. 54).

[149] Mohammadhossein Ghahramani, Yan Qiao, MengChu Zhou, Adrian OHagan and James Sweeney. *AI-based Modeling and Data-driven Evaluation for Smart Manufacturing Processes*. Aug. 2020. arXiv: `2008.12987 [cs, stat]` (cit. on p. 57).

[150] Kai Zhong, Min Han and Bing Han. 'Data-Driven Based Fault Prognosis for Industrial Systems: A Concise Overview'. In: *IEEE/CAA Journal of Automatica Sinica* 7.2 (Mar. 2020), pp. 330–345 (cit. on p. 57).

[151] H.K. Tonshoff, Xiaoli Li and C. Lapp. 'Application of Fast Haar Transform and Concurrent Learning to Tool-Breakage Detection in Milling'. In: *IEEE/ASME Transactions on Mechatronics* 8.3 (Sept. 2003), pp. 414–417 (cit. on pp. 62, 78).

[152] Guangda Xu, Huicheng Zhou and Jihong Chen. 'CNC Internal Data Based Incremental Cost-Sensitive Support Vector Machine Method for Tool Breakage Monitoring in End Milling'. In: *Engineering Applications of Artificial Intelligence* 74 (Sept. 2018), pp. 90–103 (cit. on pp. 62, 63).

[153] Guang Li, Yan Fu, Duanbing Chen, Lulu Shi and Junlin Zhou. 'Deep Anomaly Detection for CNC Machine Cutting Tool Using Spindle Current Signals'. In: *Sensors* 20.17 (Aug. 2020), p. 4896 (cit. on pp. 62, 63, 78).

[154] Sohyung Cho, Shihab Asfour, Arzu Onar and Nandita Kaundinya. 'Tool Breakage Detection Using Support Vector Machine Learning in a Milling Process'. In: *International Journal of Machine Tools and Manufacture* 45.3 (Mar. 2005), pp. 241–249 (cit. on p. 62).

[155] Sebastian Bombiski, Joanna Kossakowska and Krzysztof Jemielniak. 'Detection of Accelerated Tool Wear in Turning'. In: *Mechanical Systems and Signal Processing* 162 (Jan. 2022), p. 108021 (cit. on pp. 62, 63).

[156] Hongrui Cao, Xuefeng Chen, Yanyang Zi et al. 'End Milling Tool Breakage Detection Using Lifting Scheme and Mahalanobis Distance'. In: *International Journal of Machine Tools and Manufacture* 48.2 (Feb. 2008), pp. 141–151 (cit. on pp. 62, 78).

[157] Zepeng Liu, Zi-Qiang Lang, Yun-Peng Zhu et al. 'Sensor Data Modeling and Model Frequency Analysis for Detecting Cutting Tool Anomalies in Machining'. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 53.5 (May 2023), pp. 2641–2653 (cit. on p. 62).

[158] Gang Wang and Feng Zhang. 'A Sequence-to-Sequence Model With Attention and Monotonicity Loss for Tool Wear Monitoring and Prediction'. In: *IEEE Transactions on Instrumentation and Measurement* 70 (2021), pp. 1–11 (cit. on p. 62).

[159] Eraldo Pomponi and Alexei Vinogradov. 'A Real-Time Approach to Acoustic Emission Clustering'. In: *Mechanical Systems and Signal Processing* 40.2 (Nov. 2013), pp. 791–804 (cit. on p. 66).

[160] Santiago Velasco-Forero, Marcus Chen, Alvina Goh and Sze Kim Pang. 'Comparative Analysis of Covariance Matrix Estimation for Anomaly Detection in Hyperspectral Images'. In: *IEEE Journal of Selected Topics in Signal Processing* 9.6 (Sept. 2015), pp. 1061–1073 (cit. on p. 68).

[161] Sachin Kumar, Tommy W. S. Chow and Michael Pecht. 'Approach to Fault Identification for Electronic Products Using Mahalanobis Distance'. In: *IEEE Transactions on Instrumentation and Measurement* 59.8 (Aug. 2010), pp. 2055–2064 (cit. on p. 69).

[162] K. Worden, G. Manson and N.R.J. Fieller. 'DAMAGE DETECTION USING OUTLIER ANALYSIS'. In: *Journal of Sound and Vibration* 229.3 (Jan. 2000), pp. 647–667 (cit. on pp. 69, 70, 77).

[163] Olivier Ledoit and Michael Wolf. 'Honey, I Shrunk the Sample Covariance Matrix'. In: *UPF economics and business working paper* 691 (June 2003) (cit. on p. 70).

[164] Theanh Nguyen, Tommy HT Chan and David P Thambiratnam. 'Controlled Monte Carlo Data Generation for Statistical Damage Identification Employing Mahalanobis Squared Distance'. In: *Structural Health Monitoring* 13.4 (July 2014), pp. 461–472 (cit. on p. 75).

[165] Bernhard Scholkopf, Robert Williamson, Alex Smola, John Shawe-Taylor and John Platt. 'Support Vector Method for Novelty Detection'. In: *Proceedings of the 12th International Conference on Neural Information Processing Systems*. NIPS'99. Denver, CO: MIT Press, 1999, pp. 582–588 (cit. on p. 78).

[166] Zheng Li, Yue Zhao, Xiyang Hu et al. 'ECOD: Unsupervised Outlier Detection Using Empirical Cumulative Distribution Functions'. In: *IEEE Transactions on Knowledge and Data Engineering* (2022), pp. 1–1 (cit. on p. 78).

[167] Shen Yin, Guang Wang and Huijun Gao. 'Data-Driven Process Monitoring Based on Modified Orthogonal Projections to Latent Structures'. In: *IEEE Transactions on Control Systems Technology* 24.4 (July 2016), pp. 1480–1487 (cit. on p. 86).

[168] Junghui Chen and Kun-Chih Liu. 'On-Line Batch Process Monitoring Using Dynamic PCA and Dynamic PLS Models'. In: *Chemical Engineering Science* 57.1 (Jan. 2002), pp. 63–75 (cit. on p. 86).

[169] Weihua Li, H. Henry Yue, Sergio Valle-Cervantes and S. Joe Qin. 'Recursive PCA for Adaptive Process Monitoring'. In: *Journal of Process Control* 10.5 (Oct. 2000), pp. 471–486 (cit. on pp. 86, 90, 102, 103).

[170] Jun Shang, Maoyin Chen, Hongquan Ji et al. 'Dominant Trend Based Logistic Regression for Fault Diagnosis in Nonstationary Processes'. In: *Control Engineering Practice* (2017) (cit. on p. 86).

[171] C. K. Mechefske and D. Plummer. 'Gradual Deterioration Trending and Fault Diagnosis in Cutting Tools Using Inductive Inference Classification'. In: *International Journal of Machine Tools and Manufacture* 34.4 (May 1994), pp. 591–601 (cit. on p. 86).

[172] Neal B. Gallagher, Barry M. Wise, Stephanie Watts Butler, Daniel D. White and Gabriel G. Barna. 'Development and Benchmarking of Multivariate Statistical Process Control Tools for a Semiconductor Etch Process: Improving Robustness through Model Updating'. In: *IFAC Proceedings Volumes* 30.9 (June 1997), pp. 79–84 (cit. on p. 86).

[173] S. Joe Qin. 'Recursive PLS Algorithms for Adaptive Data Modeling'. In: *Computers & Chemical Engineering* 22.4-5 (Jan. 1998), pp. 503–514 (cit. on p. 86).

[174] Wanke Yu, Chunhui Zhao and Biao Huang. 'Stationary Subspace Analysis-Based Hierarchical Model for Batch Processes Monitoring'. In: *IEEE Transactions on Control Systems Technology* 29.1 (Jan. 2021), pp. 444–453 (cit. on p. 86).

[175] Shumei Zhang and Chunhui Zhao. 'Slow-Feature-Analysis-Based Batch Process Monitoring With Comprehensive Interpretation of Operation Condition Deviation and Dynamic Anomaly'. In: *IEEE Transactions on Industrial Electronics* 66.5 (May 2019), pp. 3773–3783 (cit. on pp. 86, 87, 90, 102).

[176] Qian Chen, Uwe Kruger and Andrew Y. T. Leung. 'Cointegration Testing Method for Monitoring Nonstationary Processes'. In: *Industrial & Engineering Chemistry Research* 48.7 (Apr. 2009), pp. 3533–3543 (cit. on pp. 86, 103).

[177] Shumei Zhang, Chunhui Zhao and Furong Gao. 'Incipient Fault Detection for Multiphase Batch Processes With Limited Batches'. In: *IEEE Transactions on Control Systems Technology* 27.1 (Jan. 2019), pp. 103–117 (cit. on pp. 86, 87, 90, 102).

[178] M. Grasso, P. Albertelli and B.M. Colosimo. 'An Adaptive SPC Approach for Multi-sensor Fusion and Monitoring of Time-varying Processes'. In: *Procedia CIRP* 12 (2013), pp. 61–66 (cit. on pp. 86, 92).

[179] Douglas C. Montgomery. *Introduction to Statistical Quality Control*. Eighth edition. Hoboken, NJ: John Wiley & Sons, Inc, 2020 (cit. on p. 87).

[180] Guang Li, Jie Wang, Jing Liang and Caitong Yue. 'Application of Sliding Nest Window Control Chart in Data Stream Anomaly Detection'. In: *Symmetry* 10.4 (Apr. 2018), p. 113 (cit. on p. 87).

[181] Xun Wang, Uwe Kruger and Barry Lennox. 'Recursive Partial Least Squares Algorithms for Monitoring Complex Industrial Processes'. In: *Control Engineering Practice* 11.6 (June 2003), pp. 613–632 (cit. on p. 87).

[182] Xun Wang, Uwe Kruger and George W. Irwin. 'Process Monitoring Approach Using Fast Moving Window PCA'. In: *Industrial & Engineering Chemistry Research* 44.15 (July 2005), pp. 5691–5702 (cit. on p. 87).

[183] John D. Healy. 'A Note on Multivariate CUSUM Procedures'. In: *Technometrics* 29.4 (Nov. 1987), pp. 409–412 (cit. on p. 87).

[184] S. Chandrasekaran, J.R. English and R.L. Disney. 'Modeling and Analysis of EWMA Control Schemes with Variance-Adjusted Control Limits'. In: *IIE Transactions* 27.3 (June 1995), pp. 282–290 (cit. on p. 87).

[185] William H. Woodall and Frederick W. Faltin. 'Rethinking Control Chart Design and Evaluation'. In: *Quality Engineering* 31.4 (Oct. 2019), pp. 596–605 (cit. on p. 88).

[186] Sang Wook Choi, Elaine B. Martin, A. Julian Morris and In-Beum Lee. 'Adaptive Multivariate Statistical Process Control for Monitoring Time-Varying Processes'. In: *Industrial & Engineering Chemistry Research* 45.9 (Apr. 2006), pp. 3108–3118 (cit. on pp. 88, 92).

[187] Chunhua Yang, Jiaojiao Zhang, Dehao Wu, Keke Huang and Weihua Gui. 'Variable Partition Based Parallel Dictionary Learning for Linearity and Nonlinearity Coexisting Dynamic Process Monitoring'. In: *Control Engineering Practice* 142 (Jan. 2024), p. 105750 (cit. on p. 88).

[188] Shumei Zhang and Xiaoli Bao. 'Phase Partition and Online Monitoring for Batch Processes Based on Harris Hawks Optimization'. In: *Control Engineering Practice* 138 (Sept. 2023), p. 105554 (cit. on p. 89).

[189] Paul Nomikos and John F. MacGregor. 'Monitoring Batch Processes Using Multiway Principal Component Analysis'. In: *AIChE Journal* 40.8 (1994), pp. 1361–1375 (cit. on p. 90).

[190] Uwe Helmke and Joachim Rosenthal. ‘Eigenvalue Inequalities and Schubert Calculus’. In: *Mathematische Nachrichten* 171.1 (1995), pp. 207–225 (cit. on p. 98).

[191] David L Donoho and Iain M Johnstone. ‘Ideal Spatial Adaptation by Wavelet Shrinkage’. In: *Biometrika* 81.3 (Sept. 1994), pp. 425–455 (cit. on p. 99).

[192] Feng Ye, Yiming Guo, Zhijie Xia, Zhisheng Zhang and Yifan Zhou. ‘Feature Extraction and Process Monitoring of Multi-Channel Data in a Forging Process via Sensor Fusion’. In: *International Journal of Computer Integrated Manufacturing* 34.1 (Jan. 2021), pp. 95–109 (cit. on p. 99).

[193] Chao Shang, Fan Yang, Xinqing Gao et al. ‘Concurrent Monitoring of Operating Condition Deviations and Process Dynamics Anomalies with Slow Feature Analysis’. In: *AIChE Journal* 61.11 (2015), pp. 3666–3682 (cit. on p. 103).

[194] Wo Jae Lee. ‘Monitoring of a Machining Process Using Kernel Principal Component Analysis and Kernel Density Estimation’. In: *Journal of Intelligent Manufacturing* 31 (2020), pp. 1175–1189 (cit. on p. 103).

[195] Kyungil Kong, Kirsten Dyer, Christopher Payne, Ian Hamerton and Paul M. Weaver. ‘Progress and Trends in Damage Detection Methods, Maintenance, and Data-driven Monitoring of Wind Turbine Blades – A Review’. In: *Renewable Energy Focus* 44 (Mar. 2023), pp. 390–412 (cit. on p. 112).

[196] Daoguang Yang, Hamid Reza Karimi and Marek Pawelczyk. ‘A New Intelligent Fault Diagnosis Framework for Rotating Machinery Based on Deep Transfer Reinforcement Learning’. In: *Control Engineering Practice* 134 (May 2023), p. 105475 (cit. on p. 112).

[197] Alessandra Caggiano, Roberta Angelone, Francesco Napolitano, Luigi Nele and Roberto Teti. ‘Dimensionality Reduction of Sensorial Features by Principal Component Analysis for ANN Machine Learning in Tool Condition Monitoring of CFRP Drilling’. In: *Procedia CIRP* 78 (2018), pp. 307–312 (cit. on p. 112).

[198] Jie Yang, Jinyong Wang, Qiaolin Ye et al. ‘A Novel Fault Detection Framework Integrated with Variable Importance Analysis for Quality-Related Nonlinear Process Monitoring’. In: *Control Engineering Practice* 141 (Dec. 2023), p. 105733 (cit. on p. 112).

[199] Yi Yang, Zhigang Ma, Alexander G. Hauptmann and Nicu Sebe. ‘Feature Selection for Multimedia Analysis by Sharing Information Among Multiple Tasks’. In: *IEEE Transactions on Multimedia* 15.3 (Apr. 2013), pp. 661–669 (cit. on p. 112).

[200] Mohammadhamed Ardakani, Mahdieh Askarian, Ahmed Shokry et al. ‘Optimal Features Selection for Designing a Fault Diagnosis System’. In: *Computer Aided Chemical Engineering*. Vol. 38. Elsevier, 2016, pp. 1111–1116 (cit. on p. 112).

[201] Minhao Zhang, Yuan Cao, Yongkui Sun and Shuai Su. ‘Vibration Signal-Based Defect Detection Method for Railway Signal Relay Using Parameter-Optimized VMD and Ensemble Feature Selection’. In: *Control Engineering Practice* 139 (Oct. 2023), p. 105630 (cit. on p. 112).

[202] Sheng Dai. ‘Variable Selection in Convex Quantile Regression: L1-norm or L0-norm Regularization’. In: *European Journal of Operational Research* 305.1 (Feb. 2023), pp. 338–355 (cit. on p. 112).

[203] Zhengxin Li, Feiping Nie, Jintang Bian, Danyang Wu and Xuelong Li. ‘Sparse PCA via L2,p-Norm Regularization for Unsupervised Feature Selection’. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021), pp. 1–1 (cit. on p. 112).

[204] Ariadna Quattoni, Xavier Carreras, Michael Collins and Trevor Darrell. ‘An Efficient Projection for $l_{1,\infty}$ Regularization’. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. Montreal Quebec Canada: ACM, June 2009, pp. 857–864 (cit. on p. 112).

[205] S. Chen, S. A. Billings and W. Luo. ‘Orthogonal Least Squares Methods and Their Application to Non-Linear System Identification’. In: *International Journal of Control* 50.5 (Nov. 1989), pp. 1873–1896 (cit. on pp. 112, 126).

[206] Xiaoquan Tang, Long Zhang and Xiaolin Wang. ‘Sparse Augmented Lagrangian Algorithm for System Identification’. In: *Neurocomputing* 330 (Feb. 2019), pp. 403–411 (cit. on pp. 113, 124, 126, 128, 139).

[207] Omid Geramifard, Jian-Xin Xu, Jun-Hong Zhou, Xiang Li and Oon Peen Gan. ‘Feature Selection for Tool Wear Monitoring: A Comparative Study’. In: *2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. July 2012, pp. 1230–1235 (cit. on pp. 113, 126).

[208] Manya V. Afonso, José M. Bioucas-Dias and Mário A. T. Figueiredo. ‘Fast Image Recovery Using Variable Splitting and Constrained Optimization’. In: *IEEE Transactions on Image Processing* 19.9 (Sept. 2010), pp. 2345–2356 (cit. on pp. 113, 125).

[209] Wei Pan, Ye Yuan, Jorge Gonçalves and Guy-Bart Stan. ‘A Sparse Bayesian Approach to the Identification of Nonlinear State-Space Systems’. In: *IEEE Transactions on Automatic Control* 61.1 (Jan. 2016), pp. 182–187 (cit. on pp. 113, 126, 127).

[210] Xiaoquan Tang, Long Zhang and Xiuting Li. ‘Bayesian Augmented Lagrangian Algorithm for System Identification’. In: *Systems & Control Letters* 120 (Oct. 2018), pp. 9–16 (cit. on pp. 113, 124, 126, 127).

[211] David J. C. MacKay. ‘Bayesian Interpolation’. In: *Neural Computation* 4.3 (May 1992), pp. 415–447 (cit. on pp. 114, 119, 125).

[212] Stephen Boyd. 'Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers'. In: *Foundations and Trends® in Machine Learning* 3.1 (2010), pp. 1–122 (cit. on p. 115).

[213] Wei Pan, Aivar Sootla and Guy-Bart Stan. 'Distributed Reconstruction of Nonlinear Networks: An ADMM Approach'. In: *IFAC Proceedings Volumes* 47.3 (2014), pp. 3208–3213 (cit. on p. 116).

[214] Michael E Tipping. 'Sparse Bayesian Learning and the Relevance Vector Machine'. In: *Journal of machine learning research* 1.Jun (2001), pp. 211–244 (cit. on pp. 119, 125, 126, 139, 141).

[215] Netti Herawati, Khoirin Nisa, Eri Setiawan, Nusyirwan Nusyirwan and Tiryono Tiryono. 'Regularized Multiple Regression Methods to Deal with Severe Multicollinearity'. In: *International Journal of Statistics and Applications* 8.4 (July 2018), pp. 167–172 (cit. on p. 122).

[216] Jonathan Eckstein and Dimitri P. Bertsekas. 'On the Douglas—Rachford Splitting Method and the Proximal Point Algorithm for Maximal Monotone Operators'. In: *Mathematical Programming* 55.1-3 (Apr. 1992), pp. 293–318 (cit. on p. 124).

[217] S. A. Billings. *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*. Chichester, West Sussex, United Kingdom: John Wiley & Sons, Inc, 2013 (cit. on pp. 125, 135).

[218] Zhe Li, Xun Wang and Uwe Kruger. 'Efficient Cross-Validatory Algorithm for Identifying Dynamic Nonlinear Process Models'. In: *Control Engineering Practice* 111 (June 2021), p. 104787 (cit. on p. 127).