©2019

ZHE GAO

ALL RIGHTS RESERVED

STATISTICAL LEARNING FOR LIGHTWEIGHT MATERIALS MANUFACTURING

by

ZHE GAO

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Industrial and Systems Engineering

Written under the direction of

Weihong Guo

And approved by

_____

_____

_____

_____

_____

New Brunswick, New Jersey

OCTOBER, 2019

ABSTRACT OF THE DISSERTATION

STATISTICAL LEARNING FOR LIGHTWEIGHT MATERIALS MANUFACTURING

By ZHE GAO

Dissertation Director:

Dr. Weihong (Grace) Guo

Lightweight materials such as Carbon fiber-reinforced polymer (CFRP) composites and aluminum alloys have been increasingly used in many industries for their high strength-to-weight ratio. However, due to their high cost, many industries, especially the automobile sector, usually mix dissimilar materials such as joining CFRP with aluminum or magnesium alloys. Friction stir blind riveting (FSBR) and micro friction stir welding (μFSW) are two solid state joining processes that have been recently extended to or developed for joining dissimilar lightweight materials, including CFRP and aluminum alloys. The overarching goal of this dissertation research is to contribute to these two advance joining techniques, FSBR and μFSW, from a process analytics perspective. For

FSBR, the specific objective is to develop new methods in sensor fusion and process monitoring to enable *in-situ* non-destructive evaluation (NDE). For μFSW, the specific objective is to develop a new method to characterize and monitor the dynamic tool wear propagation in μFSW.

In the first part of this dissertation, a systematic data-driven approach is developed for monitoring FSBR and evaluating the quality of CFRP joints using *in-situ* sensor signals. The proposed method extracts hidden information from the multi-sensor, high-dimensional, heterogeneous *in-situ* signals by unsupervised tensor decomposition; informative features are selected by sparse group lasso; the process is then monitored in real-time through the selected features by classifier fusion. A case study with FSBR experiments demonstrates the effectiveness of the proposed method in real-time evaluation of CFRP joint quality.

Since the number of training samples is usually limited in manufacturing experiments due to time and cost, the second part of this dissertation focuses on addressing the small sample size problem in data-driven quality evaluation. A novel *in-situ* NDE method is developed by integrating tensor decomposition and ensemble learning. Regularized supervised tensor decomposition extracts discriminant features from *in-situ* signals; ensemble learning is adopted to stabilize the tensor decomposition and provide better quality evaluation results. The proposed method optimizes the integration of tensor decomposition and ensemble learning by developing a novel diversity-based feature generation and selection approach: a diversity measure is defined to evaluate the extracted features; a heuristic adaptive algorithm is developed to determine the optimal parameters for integration; optimal features are selected via

clustering to maximize the diversity measure, which is expected to strengthen ensemble learning performance. Numerical studies and case study are performed to demonstrate the superiority of the proposed method over existing methods.

Tool condition in manufacturing plays a significant role on process dynamics and part quality. Effective modeling and monitoring of tool condition deterioration can provide the technical basis for maintaining production efficiency and quality. Inspired by the need of tool condition monitoring in joining processes, the third part of this dissertation aims to model and monitor the spatial and temporal patterns in μFSW tool surface measurements. A hybrid hierarchical spatio-temporal model is developed for the time-ordered, high-dimensional tool surface measurement images to characterize the dynamic tool wear propagation in μFSW. The model is developed in a hierarchical Bayesian structure with the first level being a data-driven regression model for the high-resolution tool pin profile images and the second level being a physics-based advection-diffusion model for the welding temperature distribution. Kalman filter is adopted to estimate the posterior distributions of the state variable (temperature distribution) and the error between the measured tool surface image and the predicted images. Regularized Mahalanobis distance is proposed to monitor tool wear progression. Numerical studies on three abnormal tool wear progression patterns demonstrate the effectiveness of the proposed spatio-temporal modeling method, as well as the timeliness, confidence, and power of detection.

The methodological development in this dissertation is expected to enable near real-time non-destructive evaluation of product quality, facilitate early detection of abnormal tool wear progressions, reduce the efforts in manual inspection, and support

sustainable advanced manufacturing. The proposed methods can be easily extended to other manufacturing processes with online sensing and tool measurement capabilities.

Table of Contents

List of Tables

List of Figures

CHAPTER 1 INTRODUCTION

1.1 Motivation and Background

Lightweight materials such as Carbon fiber-reinforced polymer (CFRP) composites and aluminum alloy (AA)1100-O have been increasingly used in many industries for their excellent mechanical properties comparing to other materials (Gregory and Roebroeks, 1991; Immarigeon *et al*., 1995; Gao *et al*., 2009). According to a recent McKinsey report titled "Lightweight, heavy impact", the use of lightweight materials in automotive, aviation, and wind will increase significantly in volume over the next two decades. The use of lightweight materials in automotive production will approach the levels currently used in aviation by 2030.

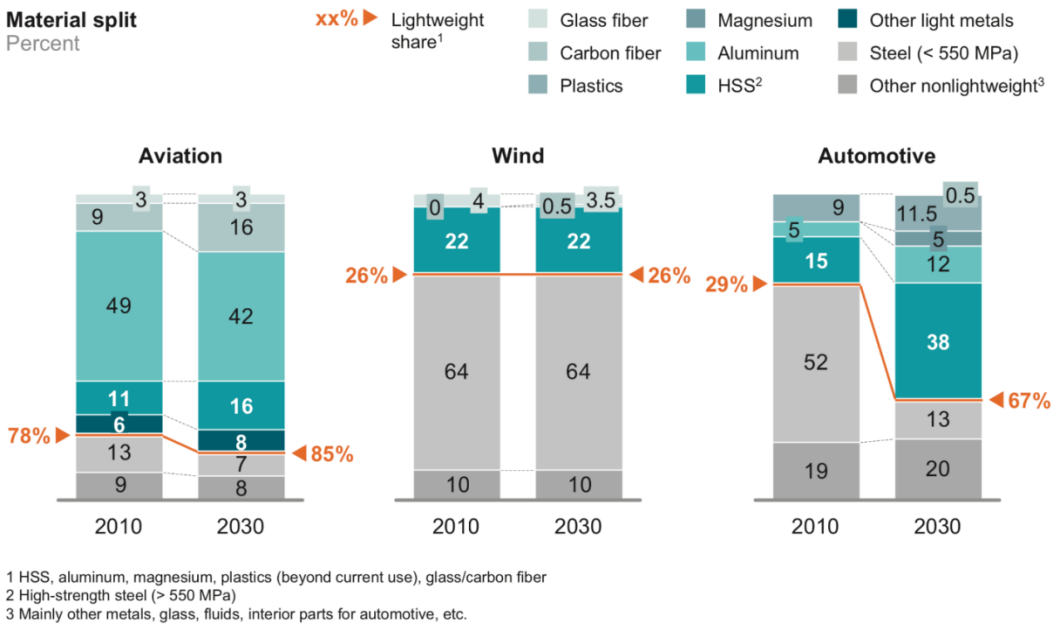


Figure 1.1 The current usage and the usage growth of lightweight materials in industry of Aviation, Wind and Automotive. Source: McKinsey

Figure 1.1 shows the global market growth of CFRP in major industry sectors since 2013. According to Lux Research report "Carbon Fiber Composites Market Update", the CFRP market will reach $35 billion in 2020. The global automotive consumption of CFRP is shown in Figure 1.2. Figure 1.3 shows the Europe aluminum alloys market size by end-user between 2013 and 2024. According to Global Market Insights, the aluminum alloys market size was registered at over $100 billion by 2016 and will reach over $150 billion after 2024.

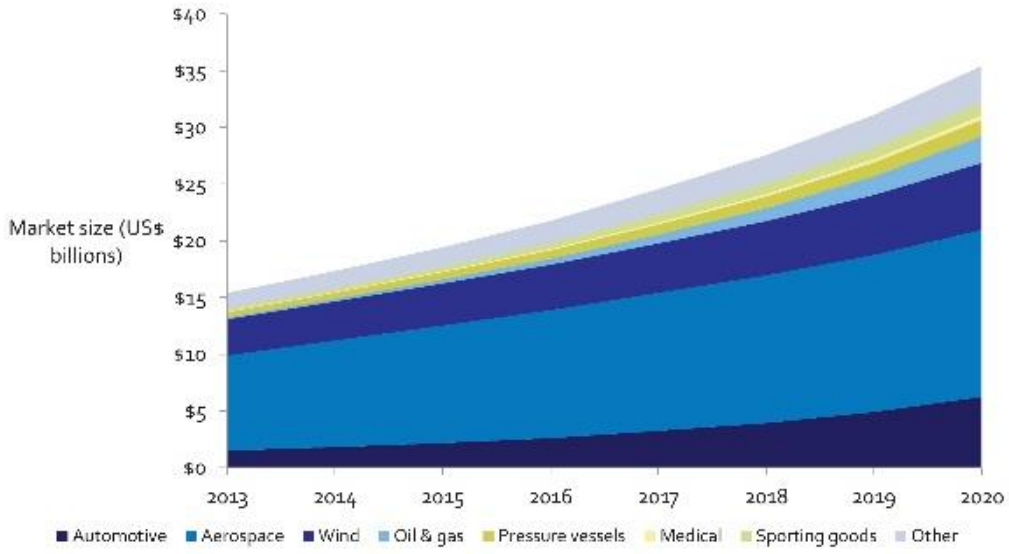


Figure 2.2 Market growth of CFRP in major industry sectors (Machine Design, http://www.machinedesign.com/contributing-technical-experts/will-carbon-fiber-find-widespread-use-automotive-industry)

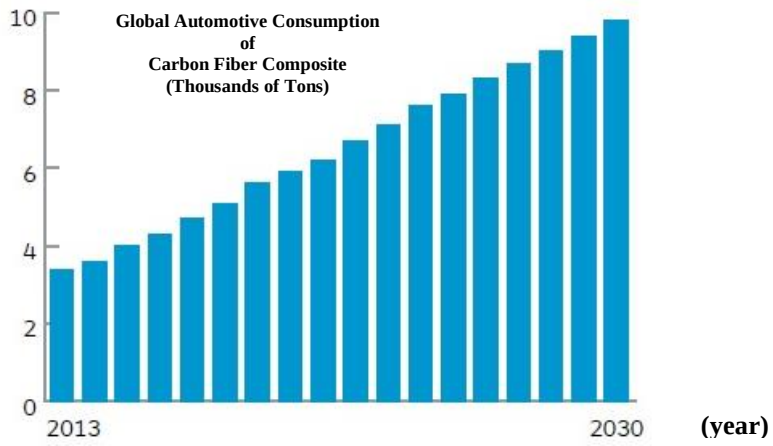


Figure 1.3 Global automotive consumption of CFRP (*Plastics Today*, https://www.plasticstoday.com/automotive-and-mobility/plastics-use-vehicles-grow-75-2020-says-industry-watcher/63791493722019)

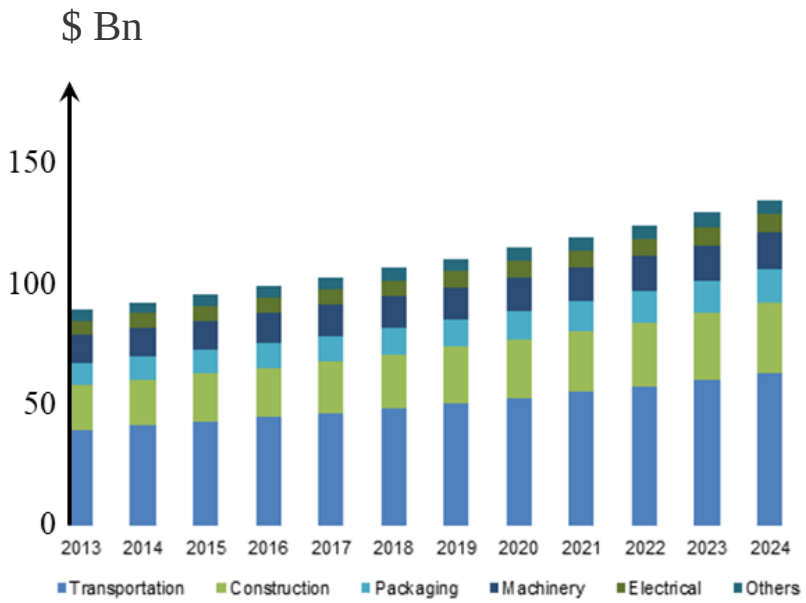


Figure 1.4 Europe aluminum alloys market size, by end-user, 2013 - 2024 ($ Bn) (Global Market Insights, https://www.gminsights.com/industry-analysis/aluminum-alloys-market)

Lightweight materials can offer significant weight advantages, but they are more expensive in comparison to the traditional materials, e.g., steel. According to the same McKinsey report, the lightweight material of aluminum weighs 60% of steel but with higher price; the lightweight material of Carbon fiber weighs half of steel but its price is

more than 5 times of the price of steel. The comparisons between the material weights and material costs are displayed in Figure 1.5.

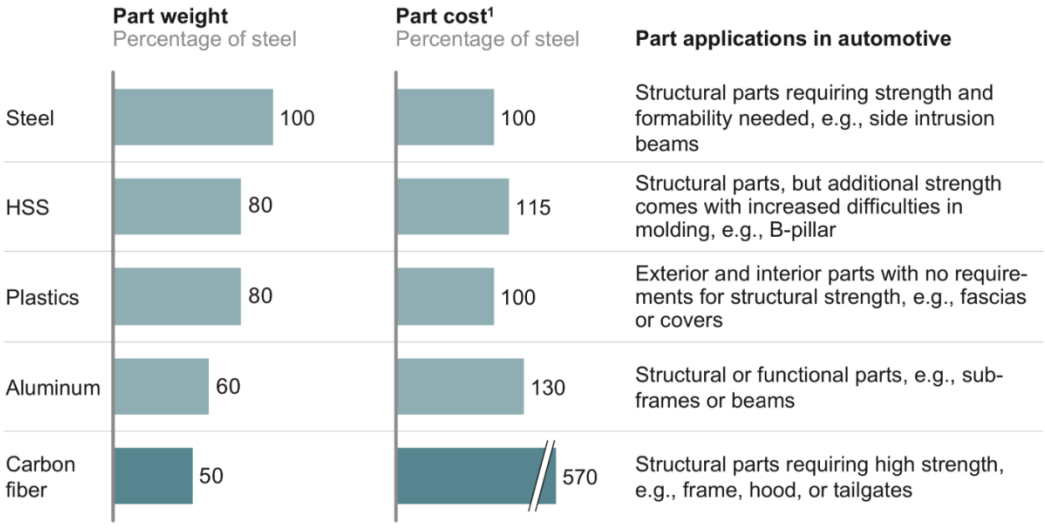


Figure 1.5 The weight and cost comparisons between lightweight materials and steel. Source: McKinsey

Since the lightweight materials are very expensive. Industry sectors usually adopt the lightweight materials by mixing them together in different lightweight packages. Shown in Figure 1.6, in the conventional lightweight package, industries replace steel with high-strength steel; in the moderate package, industries use light metals and sandwich structures; in the extreme lightweight, there is extensive usage of carbon fiber materials for maximum weight savings and this package is used for luxury cars. Due to the high cost of CFRP, many industries, especially the automobile sector, usually adopt CFRP composites by mixing CFRP with metals such as aluminum or magnesium alloys (Min *et al.*, 2015b; Min *et al.*, 2015d; Immarigeon *et al.*, 1995). Aluminum alloys in an ultra-thin format are widely used in automobile industries and they are joined with dissimilar materials that have different mechanical and physical properties, such as aluminum-steel alloys, aluminum- magnesium alloys, and aluminum-silver alloys

(Scialpi *et al*., 2008). Hence, research in joining dissimilar materials has received a lot of attention world-wide. This dissertation aims to contribute to two such joining techniques, friction stir blind riveting (FSBR) and micro friction stir welding (μFSW), from a process analytics perspective.

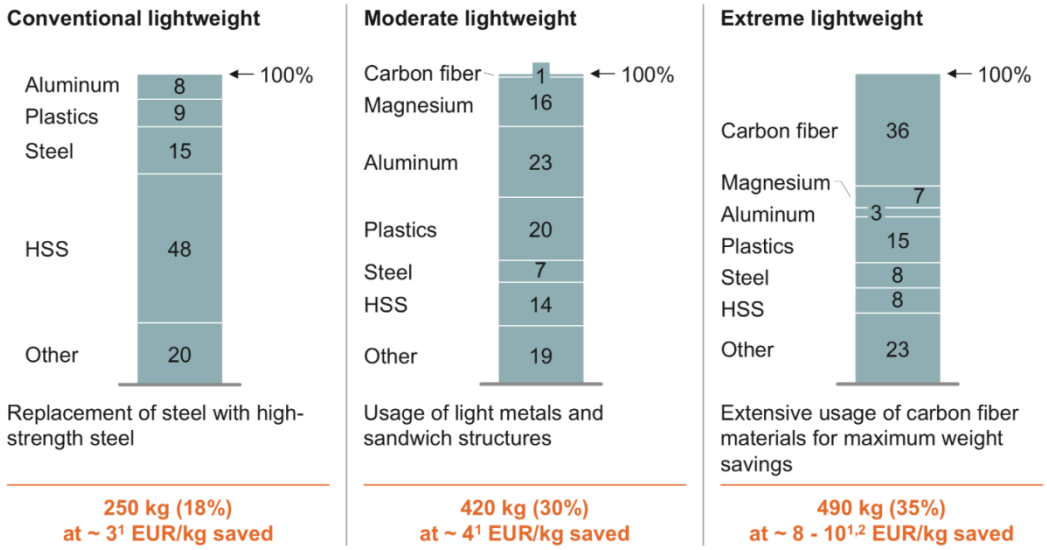


Figure 1.6 The lightweight packages. Source: McKinsey

1.1.1 Friction Stir Blind Riveting

Friction stir blind riveting (FSBR) is a joining technique that was developed to join dissimilar sheet metals. Recently, FSBR has been extended to join CFRP and aluminum alloy sheets (Min *et al*., 2015b; Min *et al*., 2015d). FSBR joins metal sheets by having a blind rivet that rotates at a high spindle speed and frictionally penetrates the workpieces. The FSBR process is illustrated in Figure 1.4: (a) the blind rivet rotates at spindle speed ω and approaches the workpieces with feed rate v; (b) the rotating blind rivet first penetrates the frictionally heated and softened top workpiece; (c) the rivet then penetrates both workpieces, leaving the shank head in contact with the top workpiece; (d) the mandrel of the blind rivet is pulled with force F; (e) the mandrel is broken and an FSBR

joint is obtained. The setup of FSBR is shown in Figure 1.5. More details of the FSBR process can be found in Min *et al*. (2015d).

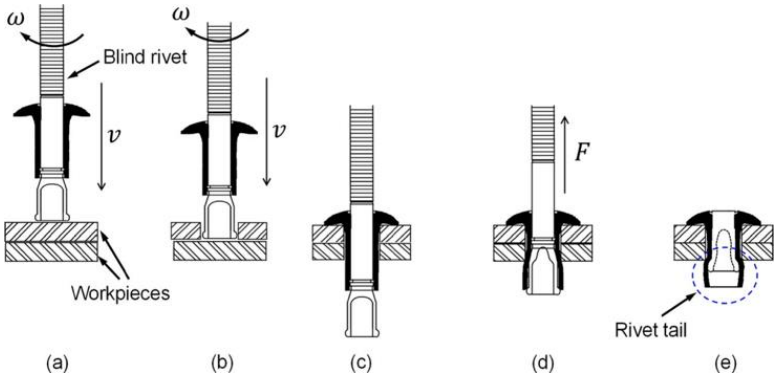


Figure 1.7 FSBR joining process (Min *et al*., 2015d)

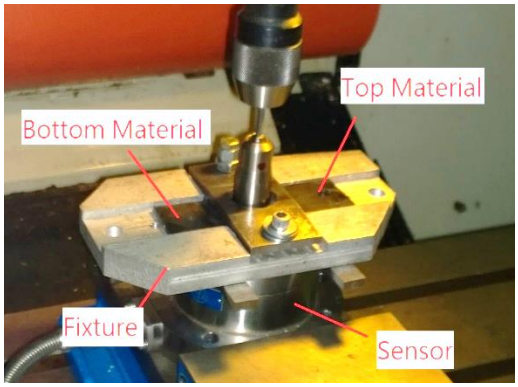
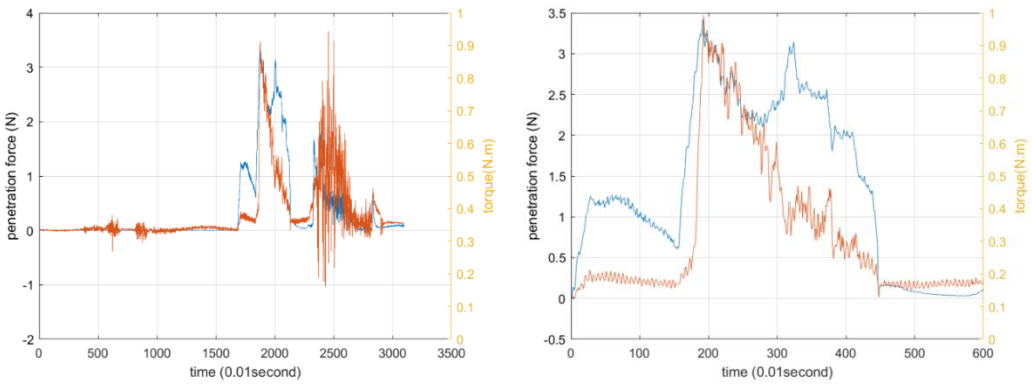


Figure 1.8 FSBR process setup (Min *et al*., 2015d)

FSBR is a one-sided mechanical joining method that takes the advantages of both friction stir welding and mechanical riveting (Lathabai *et al*., 2011). FSBR does not require predrilling a hole and thus eliminates the challenges in hole alignment (Gao *et al*., 2009). Additionally, only ~1g rivet mass is added at one joint once the internal mandrel is released. The cycle time is only a few seconds for one joint. In addition, experimental results have shown that FSBR process is generally robust in producing high strength joints (Min *et al*., 2015a; Min *et al*., 2015d; Wang *et al*., 2016).

These advantages of FSBR have attracted a lot of interest and much attention has been devoted to studying FSBR experimentally (Min *et al*., 2015a; Min *et al*., 2015d; Wang *et al*., 2016). Tensile tests were conducted to study the joint's fracture, load-extension relationship, and tensile strength. Sensors were added to the process to collect *in-situ* signals such as the penetration force and torque (Figure 1.6(a)). However, these studies only analyzed the maximum penetration force and maximum torque retrospectively, while the rich information hidden in the real-time force and torque signals were not extracted or explored. Although quality issues can be observed from tensile test results, the relationship between the FSBR process and joint quality remains unclear. Moreover, the widely used tensile test is destructive and can only be conducted offline, after the FSBR process finishes. The test may cause damages to the part while providing delayed quality information. As a result, destructive evaluations have huge negative impact on overall productivity. These limitations motivate this research to develop *in-situ* non-destructive methods for FSBR joining of lightweight materials.



(a) **(b)**

Figure 1.9 An example of the force and torque signals from FSBR: (a) raw signals (b) pre-processed signals

1.1.2 Micro Friction Stir Welding

Micro friction stir welding (μFSW) is the adaptation of the friction stir welding process (FSW) to materials with thicknesses of 1000μm or less (TWI, 2019). Friction stir welding (FSW) is a solid-state joining process wherein a non-consumable, cylindrical tool is rotated and traversed along a square butt weld joint using conventional milling techniques (Prado *et al*., 2001). FSW has enabled the joining of metals that could not be joined by other welding processes. Research has shown that dissimilar materials with very different properties, plastics, composites and even wood can be joined by FSW (Sithole and Rao, 2016). The basic concept of FSW is summarized in Figure 1.7.

μFSW further widens the applications of FSW. Applications such as thin walled structures, electrical, electronic, and micro-mechanical assemblies can benefit greatly from μFSW's ability to join a wide range of materials without the use of fluxes, shielding gases, and usually without post-weld cleaning. It is especially useful in joining dissimilar materials (Wang *et al*., 2018). A review of the developments in μFSW to date is provided in Sithole and Rao (2016). Since tool condition is very important in all manufacturing processes but has been mostly overlooked in μFSW, this research is motivated to characterize and monitor tool wear progression for μFSW.

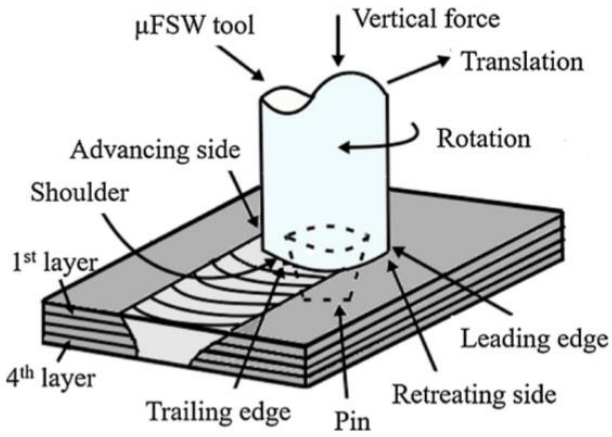


Figure 1.10 Schematic diagram of μFSW process (Wang *et al*., 2018)

1.2 Objective and Challenges

The overarching goal of this dissertation research is to contribute to two advance joining techniques, friction stir blind riveting (FSBR) and micro friction stir welding (μFSW), from a process analytics perspective. For FSBR, the research objective is to develop new methods in sensor fusion and process monitoring to enable *in-situ* non-destructive evaluation (NDE) of FSBR. For μFSW, the research objective is to develop a new method to characterize and monitor the dynamic tool wear propagation in μFSW. The specific objectives and the corresponding challenges in this dissertation are as follows:

1. *Extracting useful information from the multi-profile, high dimensional, heterogeneous in-situ signals*

In monitoring joining processes, many studies are limited to certain engineering features, such as maximum force or process duration. Although the process can be easily equipped with multiple sensors, the sensor signals are not fully utilized. There is rich information hidden in the real-time *in-situ* process signals. When multiple sensors are applied, their signals are correlated and hence should be analyzed together. In the

motivating example of FSBR, multi-profile, high dimensional, heterogeneous *in-situ* signals are captured in real-time. These signals are *heterogeneous* since different signals monitor different process variables. The signals are *high-dimensional* since a large amount of data points is collected for each sample. Calculations on the tensor format data directly would suffer from the curse of dimensionality. Therefore, informative features should be extracted from the *multi-profile, high dimensional, heterogeneous in-situ* signals.

2. *Establishing the process-quality relationship for the manufacturing process*

Establishing the relationship between process information and product quality is critical in manufacturing. Understanding this relationship enables us to monitor the process, detect process changes, estimate product quality, and all these can be done in an *in-situ* non-destructive manner. However, the process-quality relationship may be so complicated that a physics-based model cannot be well established, especially in advanced manufacturing processes or when new materials are involved. Therefore, data-driven methods are needed to establish the process-quality relationship for advanced manufacturing processes.

3. *Handling the small sample size problem*

Machine learning methods usually require a large amount of data to train the model. However, due to time and cost in conducting manufacturing experiments, usually only a small number of samples can be provided, resulting in a *small sample size* (SSS) scenario. This is not uncommon in manufacturing applications. Decision-making under SSS may lead to unreliable results. Therefore, it is important to consider the SSS problem in feature extraction, as well as how the unstable features from SSS would affect

decision-making. More importantly, a remedy for the SSS problem is in need. In this research regularization and ensemble learning will be adopted to stabilize the feature extraction process and to strengthen classification performance.

4. *Developing spatio-temporal model for µFSW tool wear progression*

Tool condition plays a significant role on process dynamics and part quality. Tool wear is complex in the sense that it varies with the combination of tool-workpiece materials and joining conditions, which makes it quite challenging to detect. Effective modeling and monitoring of tool condition degradation can provide the technical basis for maintaining production efficiency and quality. Tool surface profile measurements from different stages of the tool can be regarded as image data with both spatial and temporal dependencies. Therefore, a physics-guided spatio-temporal model is needed to characterize the dynamic tool wear progression in µFSW.

5. *Monitoring tool wear progression and detecting abnormal progression patterns*

Abnormal tool wear progression indicates that the tool condition deteriorates faster than expected, making the tool to reach its end-of-life earlier than the design life, while changing process dynamics and impairing part quality. Efficient tool wear monitoring is needed to ensure that the tool wear progression is consistent with design and to detect abnormal progression patterns. The monitoring statistics should be able to capture both spatial and temporal patterns and, at the same time, is applicable when the sample size is much smaller than the data dimension.

In summary, this research develops new methods in sensor fusion, process monitoring, and tool wear characterization and monitoring to enable *in-situ* NDE and

early detection of abnormal tool condition in lightweight materials joining. This research will establish the process-quality relationship for FSBR joining with considerations on *in-situ* sensor signals and small sample size problem; tool wear progression in μFSW will be characterized by a spatio-temporal model and monitored by regularized multivariate statistics with the potential of online implementation.

Comparing to traditional methods, the methodological development in this dissertation is expected to enable near real-time non-destructive evaluation of product quality, facilitate early detection of abnormal tool wear progressions, reduce the efforts in manual inspection, and support sustainable advanced manufacturing. The proposed methods can be easily extended to other manufacturing processes with online sensing and tool measurement capabilities.

1.3 Organization of the Dissertation

This dissertation is organized as follows. Chapter 2 provides a detailed review of related literature in quality evaluation, profile monitoring, decision-making in advanced manufacturing, and tool wear characterization and monitoring. Section 2.1 reviews the destructive and non-destructive evaluation (NDE) methods for joining processes. Both offline NDE and *in-situ* NDE methods are discussed. Section 2.2 reviews profile monitoring methods, focusing on tensor-based methods that are capable of handling multi-sensor profile data. Section 2.3 reviews decision-making methods, focusing on ensemble learning and classification. Section 2.4 reviews tool wear monitoring methods, spatio-temporal models, and the related model fitting methods. The limitations of literature are summarized in Section 2.5.

In Chapter 3, we develop an *in-situ* NDE method for joining lightweight materials using sensor fusion and process monitoring. The proposed method integrates feature extraction, feature selection, and weighted classifier fusion. We investigate the effectiveness of unsupervised tensor decomposition methods in extracting features from multi-sensor, high-dimensional, and heterogeneous profile data. The extracted features are combined with process parameters, material stack-up sequence, and engineering-driven features such as the peak force to provide better information about the FSBR process. Sparse group lasso regression is adopted to select the optimal monitoring features. The selected features are fed into weighted classifier fusion to estimate the joint quality. The fusion method integrates five individual classifiers with optimal weights. The correct classification rates resulted from various feature extraction and selection methods are assessed and compared.

In Chapter 4, we consider the limited amount of available experiment samples but all training samples have known quality information. We improve the process monitoring and quality evaluation by integrating supervised feature extraction, diversity-based optimal feature selection and ensemble learning. Multilinear extensions of linear discriminant analysis (MLDA) are adopted for supervised feature extraction. Adding regularization to MLDA ensures the tensor decomposition solution exists under the small sample size (SSS) scenario. Ensemble learning is integrated with tensor decomposition to stabilize the feature extraction and to provide more accurate quality evaluation results. The integration is optimized by selecting the optimal features to boost learner diversity and by determining the optimal regularization parameter so that the extracted features will perform the best in ensemble learning.

In Chapter 5, we investigate tool wear characterization and monitoring when tool surface measurement images from different stages of tool usage are available. Micro friction stir welding is the motivating example behind this chapter but the method development can be extended to other manufacturing processes. With a sequence of tool surface measurement images, we develop a hybrid hierarchical spatio-temporal model to characterize the dynamic tool wear propagation in μFSW. The proposed model is developed in a hierarchical Bayesian structure with the first level being a data-driven regression model for the high-resolution tool pin profile images and the second level being a physics-based advection-diffusion model for the welding temperature distribution. Kalman filter is adopted to estimate the posterior distributions of the state variable (temperature distribution) and the error between the measured tool surface image and the predicted images. Regularized Mahalanobis distances are derived for tool wear progression monitoring.

Chapter 6 concludes the dissertation and discusses future work.

CHAPTER 2 LITERATURE REVIEW

In this chapter, we present a detailed overview of research related to the problem being investigated in this dissertation. We review the related literature in quality evaluation, profile monitoring, decision-making in advanced manufacturing, tool condition monitoring, and spatio-temporal modeling. We summarize the limitations of current research and highlight the uniqueness of our research.

We first review the existing methods in quality evaluation for joining processes, including destructive methods and non-destructive methods. Focusing on non-destructive evaluation (NDE), we review *in-situ* NDE methods and offline methods. Despite the vast literature on NDE for joining techniques, *in-situ* NDE for FSBR joining of lightweight materials is limited. Since FSBR can be regarded as a combination of Friction Stir Welding (FSW) and Mechanical Riveting (MR) (Gao *et al*., 2009) and it is also similar to the FricRiveting joining technique, we also review the current quality evaluation methods for FSW, MR and FricRiveting.

A major component of *in-situ* NDE is process monitoring. With sensors are installed on the equipment, *in-situ* sensor signals are widely available. We review the current research on profile modeling and monitoring. Current research in this direction focuses on reducing data dimensionality and extracting a few representative informative features from the original signals. Since our preliminary analysis of FSBR data shows that the sensor signals are nonlinear in shape and that they come from different sensors, we focus our review on tensor decomposition methods that are capable of extracting features from multi-profile data. Most of these methods were developed for face and gait

recognition tasks, while their applications in manufacturing have been limited. No related work has been done for FSBR joining of lightweight materials.

We also review the current research on decision-making. Since our research focuses on identifying poor joints from good joints, classification applies. Weak classifiers can be strengthened by ensemble learning, so, we review the existing ensemble learning algorithms for classification. Integrating ensemble learning with tensor decomposition can potentially boost the overall classification performance, yet no research has been conducted in this area.

Since tool condition affects process dynamics and part quality, tool wear characterization and monitoring are important topics in manufacturing. Since tool wear monitoring in μFSW has not been fully studied yet, we consider the related studies for FSW. We review the related literature in tool wear monitoring, spatio-temporal modeling, and model fitting methods. Tool wear characterization considering both spatial and temporal dependences is an area that has not been fully tapped. Tool wear characterization and monitoring for μFSW haven't been reported in literature.

2.1 Quality Evaluation for Joining Processes

To evaluate the quality of a joint, the evaluation methods usually refer to the use of invasive or noninvasive techniques to (1) determine the integrity of the material, component, or structure or (2) quantitatively measure some characteristics of the joint (Center, 2014). If the testing methods are invasive, they are considered as *destructive evaluation*; otherwise, they are *non-destructive evaluation (NDE)* methods.

2.1.1 Destructive Evaluation Methods

In destructive evaluation, the joint is inspected in an invasive way after joining. The majority destructive evaluation methods focus on studying the mechanical or chemical properties of the joint. For example, to measure the strength of a joint, typical destructive evaluation methods include bend test (Threadgill *et al*., 2009; Gibson *et al*., 2014a), tensile test (Gao *et al*., 2009; Zhang *et al*., 2011; Min *et al*., 2014; Min *et al*., 2015c), and lap shear test (Blaga *et al*., 2015; Min *et al*., 2015c). To investigate the formation of a joint, fatigue test (Lee, 2005; Kahl, 2010) and scratch test (Wredenberg and Larsson, 2009) are usually used. The atmospheric corrosion test (Li *et al*., 2016) is used to test how long the FSBR joint will last under corrosion.

Destructive evaluation is essential in providing definitive conclusions for the quality of a joint, as well as findings in joint formation and microstructures (Threadgill *et al*., 2009; Gibson *et al*., 2014a). However, this evaluation usually requires a complex and time-consuming testing procedure, which can only be performed after the joining process, providing delayed quality information that cannot be used for *in-situ* process adjustments directly; the testing product after destructive evaluations usually has to be disposed, increasing the total production cost. Hence, there is a huge interest in developing non-destructive evaluation methods for joining processes.

2.1.2 Non-Destructive Evaluation (NDE) Methods

Depending on when the non-destructive evaluation is performed, NDE methods can be categorized into offline NDE, which is performed after the joining process, and *in-situ* NDE, which is performed during the process.

2.1.2.1 Offline NDE

Post-process, offline NDE has been widely used in traditional joining techniques, such as FSW, MR and FricRiveting; some of these NDE methods are gradually extended to FSBR.

Visual tests are widely used to examine the joint microstructure (Threadgill *et al*., 2009; Altmeyer *et al*., 2014; Blaga *et al*., 2015); dye penetration tests (Wu and Stepinski, 1998; Kinchen and Aldahir, 2002; Suits *et al*., 2003; Li *et al*., 2011) are conducted to locate surface-breaking defects in non-porous materials; eddy current testing (Kinchen and Aldahir, 2002; Suits *et al*., 2003) uses electromagnetic induction to detect and characterize the surface or sub-surface flaws in conductive materials; radiographic testing (Lamarre *et al*., 2000; Santos and Santos, 2010; Li *et al*., 2011; Li *et al*., 2015) measures the internal structure and the integrity of the joint materials; ultrasonic testing (Sato *et al*., 2002; Myers *et al*., 2008; Fehrenbacher *et al*., 2011; Jiang *et al*., 2011) can detect internal flaws.

Although these offline NDE methods are widely used in joining techniques, they are subject to two major drawbacks. First, even if high evaluation accuracy can be obtained, the post-process quality information is delayed comparing to *in-situ* evaluations; such delayed quality information are too late to be used for online process adjustment (Gibson *et al*., 2014a). Second, some of the offline NDEs must be operated by experts, such as eddy current testing, radiographic testing, and ultrasonic testing. The accuracy and efficiency of these procedures heavily depend on the expertise and experience of the technician. Human errors or lack of experience may have huge impact on the evaluation

results. These drawbacks point to a lack of automatic, *in-situ* non-destructive method for joint quality evaluation.

2.1.2.2 *In-situ* NDE

There is a huge interest in *in-situ* NDEs for their advantages in reducing inspection efforts, eliminating production waste, and providing immediate response. *In-situ* NDE is regarded as a complementary evaluation technique to the post-process offline NDE methods (Gibson *et al*., 2014a). With *in-situ* NDE, the joint quality can be evaluated in an automated and near real-time fashion by monitoring informative features such as process parameters and *in-situ* sensor signals. The most critical element in *in-situ* NDE is establishing the relationship between the *in-situ* process information and the quality or properties of the joint.

To better interpret the *in-situ* process information, especially sensor signals, researchers have explored various approaches including signal transformation and machine learning to reduce the data dimensionality and extract the informative features for profile monitoring. To determine the process-property relationship, classification and regression are usually used for decision-making. Effective *in-situ* NDE methods have been reported for FSW (Wang *et al*., 1982; Chen *et al*., 2003; Boldsaikhan *et al*., 2006; Boldsaikhan, 2008; Bhowmick, 2010; Boldsaikhan *et al*., 2010; Boldsaikhan *et al*., 2011), ultrasonic welding (Guo *et al*., 2016), gas metal arc welding (Saini and Floyd, 1998; Adolfsson *et al*., 1999), and rivet joining (Maurer *et al*., 2002; Ihn and Chang, 2004). Guo *et al*. (2017) is the only existing work on *in-situ* NDE for FSBR. Since our research aims at *in-situ* NDE, we present a detailed review of the related literature in profile

monitoring and decision-making in the following subsections. Drawbacks of Guo *et al*. (2017) will also be discussed.

2.2 Profile Monitoring in Advanced Manufacturing

In manufacturing, the sensor measurements provided by online sensing and data capturing technology are time- or spatial-dependent functional data, also called profile data (Paynabar *et al*., 2013; Guo *et al*., 2016). For manufacturing data, the profiles could be linear or nonlinear in shape, could come from one sensor or multiple sensors, and could be measuring one or multiple characteristics.

Monitoring linear profiles has received extensive research (Kang and Albin, 2000; Kim *et al*., 2003; Mahmoud and Woodall, 2004; Gupta *et al*., 2006; Zou *et al*., 2006; Mahmoud *et al*., 2007; Jensen *et al*., 2008; Noorossana *et al*., 2011). Although great theoretical analysis and application results were reported in these literatures, a lot of profile data in manufacturing applications are nonlinear. The preliminary analysis in Figure 1.5 shows the FSBR data are also nonlinear.

Current research on monitoring individual nonlinear profiles has been reported in Gardner *et al*. (1997), Jin and Shi (2001), Walker and Wright (2002), Jeong *et al*. (2006b), Jung et al. (2006a), Mosesova *et al*. (2006), Zou *et al*. (2008), Vaghefi *et al*. (2009), Zou *et al*. (2009), Qiu *et al*. (2010), Paynabar and Jin (2011), Paynabar *et al*. (2012) and etc. These studies are limited to individual profiles, which is the case if only one sensor signal is captured. When multiple sensors are available, these analyses would model the profiles one at a time, ignoring sensor-to-sensor correlations.

Nowadays, it’s easy and feasible to install multiple sensors to collect different variables during a manufacturing process. Our preliminary analysis shows that two

sensors are installed to FSBR to capture force and torque signals, respectively. Because of sensor-to-sensor correlation, it is important to monitor these multiple profiles together. Recent studies on multi-profile monitoring include Gao and Durrant-Whyte (1994), Jeong et al. (2006a), Jung *et al*. (2006b), Kim *et al*. (2006), Basir and Yuan (2007), Salahshoor *et al*. (2008), Lei *et al*. (2010), Quan *et al*. (2010), Paynabar *et al*. (2013), Grasso *et al*. (2014), Guo *et al*. (2016), and Jeong *et al*. (2018). The multi-profile monitoring methods developed in these studies can be summarized into parametric and non-parametric methods.

Parametric methods model the nonlinear multi-profile data in a parametric form, such as multivariate normal distribution, and then use a suitable decision-making method, such as control chart, to monitor the model parameters (Walker and Wright, 2002; Vaghefi *et al*., 2009). Parametric methods assume the profile data follow a certain distribution, but this assumption may not hold in many situations. Parameterization of nonlinear profiles may be difficult. In these situations, non-parametric approaches are preferred. Non-parametric methods model the profile data in non-parametric forms, such as spline regression (Chang and Yadama, 2010) and wavelets (Jin and Shi, 2001), followed by a decision-making process (Lucas and Saccucci, 1990; Fan and Gijbels, 1996; Fan *et al*., 2001; Wu and Zhang, 2002; Noorossana *et al*., 2011).

In this research, since our profiles are high-dimensional, direct calculation of this data can be extremely time-consuming and unreliable. Our multi-profile monitoring approach should be able to reduce data dimensionality and extract informative features from the original profiles. Linear discriminant analysis (LDA) is a widely used parametric dimension reduction and feature extraction method; principal component

analysis (PCA) is a widely used non-parametric dimension reduction and feature extraction method. The multilinear extensions of these two methods can extract informative features from the tensor format data directly, without vectorizing the profiles.

2.2.1 PCA-Based Profile Monitoring Methods

The principal component analysis (PCA) is a well understood and widely used unsupervised non-parametric multivariate technique that projects a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components (Friedman *et al*., 2001, Bilodeau and Brenner, 2008). The principal components are ranked according to the variance (energy) they occupy. Being an unsupervised method, PCA projection analyzes the variance of the original data, without requiring class information. The top principal components are able to explain a majority of the variance of the original data. By setting a percentage threshold on the variance to capture, PCA can serve as an unsupervised feature extraction method.

While PCA can only handle two-dimensional data, multilinear extensions of PCA have been developed for high-dimensional data, known as tensor objects (Lu *et al*., 2008a). These extensions project the tensor format multi-profile data from the high-dimensional tensor space directly to a low-dimensional space, without vectorizing the tensor object. The multilinear PCA methods were originally developed for face and gait recognition (Yang *et al*., 2004; Ye *et al*., 2004; Ye, 2005; Lu *et al*., 2008a), where digitally captured surveillance images or videos are projected into principal components via tensor-to-tensor or tensor-to-vector projections (Lu, 2008). A comparison of the four most widely used multilinear PCA methods is given in Table 2.1.

Recently, multilinear extensions of PCA have been gradually extended to manufacturing applications for interpreting sensor signals. For example, Grasso *et al*. (2014) monitored the multichannel signals for waterjet cutting process with MPCA and VPCA methods; Paynabar *et al*. (2013) undertook UMPCA to extract uncorrelated features for multi-channel signals in tonnage process; Yan *et al*. (2015) applied four multilinear PCA methods to steel tube manufacturing and compared their performance in detecting abnormal furnace combustions. However, no PCA-based research has been conducted on monitoring signals from the FSBR process.

Table 2.1 Multilinear extension of PCA methods (Adapted from Lu *et al*. (2013))

Algorithm	***Projection Method***	***Objective Function***	***Notes***
MPCA	Tensor-to-tensor projection	Variance maximization	Extracted features may be correlated.
TROD	Tensor-to-vector projection	Least square error minimization	Input tensor must be decomposed into linear combinations of rank-one tensors.
UMPCA		Variance maximization	Extracted features are uncorrelated.
VPCA	Unfold the tensor into lower dimensional linear arrays	Variance maximization	Input tensor must be unfolded to a matrix and then apply the regular PCA on the matrix. But the original data structure is broken and information may be lost after feature extraction

2.2.2 LDA-Based Profile Monitoring Methods

Linear discriminant analysis (LDA) is another widely used dimensionality reduction method. As a supervised method, LDA takes the class information into account and extracts features to maximize the differences among classes. Thus, LDA-based methods are preferred if class information is available.

The original LDA algorithm works only for vector data on the two-dimensional space and later it has been extended to high-dimensional space. The multilinear extension of LDA-based methods is named as the multilinear discriminant analysis (MLDA) in general. MLDA projects tensor data from high-dimensional space to a low-dimensional space by maximizing the scatter ratio. The scatter ratio is the ratio between the between-class scatter and the within-class scatter. The MLDA methods were developed for face and gait recognition (Lu *et al*., 2013). Differently from the PCA-based methods, LDA-based methods may not work very well with small training sample size. This is called small sample size (SSS) scenario, under which the within-class scatter matrix may always be singular. Regularization is usually used to handle the SSS issue (Friedman, 1989).

Table 2.2 summarizes the most widely used multilinear extensions of LDA. 2DLDA (Two-Dimensional LDA) was developed in Ye *et al*. (2005) to apply LDA on high-dimensional data; different from the LDA which only operates the vectorized data, the 2DLDA launches the calculation on matrices. Yan *et al*. (2007) extended 2DLDA for higher-dimensional tensor data and the new method was called DATER (discriminant analysis with tensor representation). Later, GTER (general tensor discriminant analysis), a deviation of DATER, was introduced in Tao *et al*. (2007) by changing the original scatter ratio criterion to scatter difference criterion which optimizes the difference

between the between-class scatter matrix and the within-class scatter matrix. 2DLDA, DATER, and GTER are based on tensor-to-tensor projection (TTP), which maps the input tensor data into another multilinear array of the same order (Lu *et al*., 2013). The other type of projection is tensor-to-vector projection (TVP), which maps the input data from the tensor space into vectors (Lu *et al*., 2013). TR1DA (Tensor rank one discriminant analysis) is a TVP algorithm developed in Wang and Gong (2006) to obtain a series of rank one tensors by optimizing the scatter difference criterion from the repeatedly calculated residuals. R-UMLDA, developed in Lu *et al*. (2009), is another MLDA extension based on tensor-to-vector projection. R-UMLDA combines regularized-LDA (RLDA) and uncorrelated-LDA (ULDA) and extends this combination from the two-dimensional space to high-dimensional tensor space to extract uncorrelated features. The regularization in R-UMLDA provides a remedy for the SSS issue. The aforementioned methods were mostly developed for face and gait recognition (Yan *et al*., 2007; Lu, 2008). Recently, LDA-based methods have been extended to manufacturing applications. Guo *et al*. (2016) adopted R-UMLDA to investigate the fault diagnosis in multi-layer ultrasonic metal welding and showed it is promising to use UMLDA in manufacturing. However, LDA-based methods have not been utilized for investigating FSBR or lightweight material joining.

Table 2.2 Multilinear discriminant analysis algorithms (Adapted from Lu *et al*. (2013))

Algorithm	***Projection Method***	***Discriminant Criterion***	***Notes***
2DLDA	Tensor-to-tensor projection	Scatter ratio maximization	Replace vector input to matrix input Data dimension: $D = 2$
DATER			Data Dimension $D > 2$
GTDA		Scatter difference maximization	Data Dimension $D > 2$
UMLDA	Tensor-to-vector projection	Scatter ratio maximization	No correlation among extracted features
R-UMLDA			No correlation among extracted features and w/ regularization
TR1DA		Scatter difference maximization	Tensor rank one decomposition

2.3 Decision-Making for in-situ NDE

The quality results for *in-situ* NDE can be determined via various decision-making methods, including control charts, regression, and classification (Noorossana *et al*., 2011). In this research, since we are interested to know if a joint is in high quality or poor quality, classification applies. The classification task can be done by an individual classifier or by fusing the results from multiple classifiers. Using an individual classifier is easy and fast, but the performance may be subject to training sample size, parameters in the classifier, dimensionality of the data and etc. Therefore, using multiple classifier is recommended if these concerns cannot be addressed by an individual classifier. The most straightforward method of using multiple classifiers is (weighted) classifier fusion, in which the final results are linear combinations of the results from individual classifiers. Classifier fusion is representative of the “mixture of experts” family in ensemble learning

(Zhou, 2012), under which individual learners are trained for different subtasks and combined via certain rules to assign the outputs (Jacobs *et al*., 1991; Xu *et al*., 1995). The "mixture of experts" was initially designed for neural networks and we adopt it to evaluate the joint quality in Chapter 3. In general, ensemble learning algorithms ensemble weak learners to form a strong learner. The ensembles of weak learners may be more powerful than a classifier fusion method (Zhou, 2012).

2.3.1 Ensemble Learning

Ensemble learning algorithms can improve classification performance via a learning sequence of: (1) select the proper base learner; (2) generate the base learner committee; (3) combine the results from individual learners to obtain the final ensemble learning result. The diversity of base learner is critical in ensuring good ensemble performance. Diversity can be manipulated from different aspects during the ensemble learning sequence.

Pioneer works, such as Hansen and Salamon (1990) and Schapire (1990) demonstrated ensemble learning performance empirically and proved the ensemble learning mechanism theoretically. The performance from one ensemble learning algorithm relies on four major factors: the selection of the base learner, the construction of the base learner committee, the combination of the base learners (combining rule) and the manipulation of the base learner diversity.

A good ensemble learning algorithm requires the base learners to be accurate, diverse, but not strong (Chen, 2008). A base learner is considered *accurate* if the performance of this base learner is better than random guess; two base learners are considered *diverse* if they make different learning results on the same observation; the

base learner is considered *not strong* if its accuracy is only slightly better than random guess. Decision tree meets all three requirements very well and thus is a good choice for base learner (Friedman *et al*., 2001; Zhou, 2012).

The base learner committee can be constructed by two methods:

1. The sequential method studies the dependence among base learners and generates new base learners sequentially to reduce the learning error from the previous base learner, with the Adaboost method as a representative. Boosting method generates classifiers sequentially by multiplying weights on the training data and then takes a weighted majority vote on these classifiers to assign the final classification outputs (Schapire, 1990; Schapire *et al*., 1998; Schapire and Singer, 1999; Friedman *et al*., 2000; Schapire, 2003). The weights in the current algorithm iteration are usually the functional results of the classification error in the previous iteration and are normalized to follow a distribution. Variations of the boosting method have been developed for binary and multi-class classification tasks (Schapire *et al*., 1998; Friedman *et al*., 2000; Freund, 2001; Demiriz *et al*., 2002; Warmuth *et al*., 2006; Freund, 2009; Zhou, 2012).
2. The parallel method studies the independence among base learners and then reduces the generalization error via certain combining rules, with Bagging as a representative (Zhou, 2012). Bagging constructs the base learners independently via bootstrapped sampling (Breiman, 1996). Theoretical studies using the Hoeffding's inequality have proved that via the majority voting combining rule, the final generalization error for classification tasks

under the Bagging method can be reduced significantly (Zhou, 2012). For example, using the majority voting rule, for a binary classification task, the error on an observation from Bagging is defined only when more than half of the base learners classified this observation into the wrong class and therefore Bagging method significantly reduces classification error. Moreover, Bagging is especially suitable for unstable learners under a small training sample size, since bootstrapped sampling can enlarge the training sample size.

After generating a set of base learners, the ensemble learning resorts to combining rules to achieve the strong generalization ability. Benefits from the combining rules can be summarized as follows. Classifier combination can reduce the estimation variance, especially when dealing with limited sample size. Classifier combination can run a local search from many different starting points and reduce the possibility of getting stuck in the wrong local optimal results. Moreover, classifier combination is able to establish a more accurate functional relationship between the input data and the output classification labels (Zhou, 2012; Dietterich, 2000). Two combining rules are considered in this dissertation: the weighted averaging rule for Adaboost and the majority voting rule for Bagging.

Ensemble learning algorithms rely on the base learner diversity to provide a good learning performance. There are four major manipulation methods to generate base learner diversity: (1) data sample manipulation generates diversity from the sample space (e.g., Bagging and Adaboost); (2) input feature manipulation generates diversity from the feature space (e.g., Random Subspace (Ho, 1998)); (3) learning parameter manipulation generates diversity by input randomness on parameters, (e.g., Negative Correlation

Method (Liu and Yao, 1999)) ; and (4) output representation manipulation generates diversity from the output results backward to learning process (e.g., Error-Correcting Output Coding (ECOC) (Dietterich and Bakiri, 1994)). Specifically, both data sample manipulation and input feature manipulation are adopted in Random Forest (Breiman, 2001). A summary information of the aforementioned ensemble learning algorithms is provided in Table 2.3.

Although ensemble learning can effectively improve the classification performance for data on the two-dimensional space, limited research has been conducted on the integration of tensor decomposition and ensemble learning. No research has been done on developing an optimal integration that takes the advantages from both tensor decomposition and ensemble learning. Further, the ensemble learning algorithms under classification setting are usually proposed for pattern recognition tasks in social sciences, rather than manufacturing applications. No related research has been seen for lightweight materials manufacturing yet.

Table 2.3 Ensemble learning algorithms

Ensemble learning algorithm		***Binary (B) or multiclass (M)***	***Base learner constructed by parallel (P) or sequential (S) generation***	***Base learner combination methods***	***Diversity generation methods***	***Additional constraints***
Random Subspace		B, M	P	Majority voting	Feature manipulation	None
Negative Correlation Method		B, M	P	Simple averaging voting	Learning parameter manipulation	Design for neural network ensemble learning
Error-Correcting Output Coding		M	N/A	N/A	Output representative manipulation	None
Random Forest		B, M	P	Majority voting	Feature and sample manipulation	None
Bagging		B, M	P	Majority voting	Sample manipulation	None
Adaboost	Adaboost M1	B	S	Convex optimal weighted averaging	Sample manipulation	None
	Adaboost M2	M	S	Convex optimal weighted averaging	Sample manipulation	None
	Logitboost	B, M	S	Convex optimal weighted averaging	Sample manipulation	1.Fit the additive logistic regression method; 2.Minimize the log loss function
	Gentleboost	B	S	Convex optimal weighted averaging	Sample manipulation	1.Fit the additive logistic regression method; 2.Minimize the expectation of the exponential loss function

Adaboost	LPboost	B, M	S	Convex optimal weighted averaging	Sample manipulation	1.Maximizes the classification margin; 2. Apply linear programming algorithm
	Totalboost	B, M	S	Convex optimal weighted averaging	Sample manipulation	1.Maximizes the classification margin; 2. Apply convex programming algorithm
	Brownboost	B	S	Nonconvex optimal weighted averaging	Sample manipulation	1.Maximizes the classification margin; 2. Tolerate noise
	Robustboost	B	S	Nonconvex optimal weighted averaging	Sample manipulation	1.Maximizes the classification margin; 2. Tolerate noise
	RUSboost	B, M	S	Convex optimal weighted averaging	Sample manipulation	Apply to uneven training sample size per class

2.4 Tool Wear Characterization and Monitoring

In this section, we present the state-of-the-art related to the research problem being studied in Chapter 5. We review the related literature in tool wear monitoring, spatio-temporal modeling, and model fitting methods. We summarize the limitations of the current work and highlight the contributions of our research. Since tool wear monitoring in μFSW has not been fully studied yet, we consider the related studies for FSW and its adaptions in this section. These efforts have provided us with necessary instructions and inspired our research in Chapter 5.

2.4.1 Tool Wear Monitoring Methods

Friction stir welding (FSW) is a solid-state joining process wherein a non-consumable, cylindrical tool is rotated and traversed along a square butt weld joint using conventional milling techniques (Prado *et al*., 2001). μFSW is the adaptation of the FSW process to materials with thicknesses of 1000μm or less (TWI, 2019; Wang *et al*., 2018). In both FSW and μFSW, tool wear affects tool usage condition, tool remaining life, welding quality, process control, and manufacturing economy (Pfeifer and Wiegers, 2000; Haber *et al*., 2004; Rai *et al*., 2011; Gibson *et al*., 2014b). Therefore, a tool condition monitoring (TCM) system that can detect tool wear levels and predict tool remaining useful life can provide great benefits. Extensive research work on tool condition monitoring is taking place world-wide. It has been observed from the literature that TCM has been extensively studied for machining processes, e.g., turning, milling, drilling and etc. (Konrad *et al*., 1994; Fu *et al*., 1998a; Kurada and Bradley, 1997; Zhang *et al*., 1995; Jantunen, 2002; Kannatey-Asibu and Dornfeld, 1981; Chin *et al*., 1994; Li and Li, 1993; Diei and Dornfeld, 1987; Fu *et al*., 1998b; Pai and Rao, 2002). However, it has also been

observed that, there is little published information available about the TCM in FSW or μFSW.

TCM for machining process includes direct methods and indirect methods (Ahmad *et al*., 2015). Direct methods measure the geometric features of the machining tool directly using optical and visual approaches (Devillez *et al*., 2004). Indirect methods infer tool condition from signal sensors via signal processing, sensor fusion, computer vision, artificial intelligence and so on (Ahmad *et al*., 2015). Signals commonly used in indirect TCM include acoustic emission (Choi *et al*., 1999; Jemielniak and Otman, 1998), temperature (Chow and Wright, 1988; Boothroyd, 1988), cutting forces (static and dynamic) (Choi *et al*., 1999; Dimla Snr, 1999), vibration (Dimla Sr and Lister, 2000; El-Wardany *et al*., 1996), ultrasound (Abu-Zahra and Nayfeh, 1997), and other optical measurements (Martin *et al*., 1986).

TCM for joining processes is very limited, especially for FSW. The first reported work on TCM for lightweight dissimilar materials joining is the TCM for ultrasonic metal welding in Shao *et al*. (2014).. Later, Shao *et al*. (2016) developed TCM for ultrasonic metal welding of lithium-ion batteries. Our research in Chapter 5 will be the first investigation of TCM for μFSW.

TCM for FSW or μFSW has received little attention. In various adaptions of FSW, tool condition is not a major concern comparing to other process conditions. This could be partly ascribed to the following reasons: (1) welding tools in all adaptions of FSW are considered non-consumable and therefore, tool wear in FSW may not cause severe damages or may not happen as frequently as the tool wear in machining process (Prado *et al*., 2001); (2) empirical study shows that welding tools in FSW may develop a desirable

tool geometry after an initial wear period that can optimize the welding process (Prado *et al*., 2003; Shindo *et al*., 2002; Rai *et al*., 2011), making the tool wear in FSW desirable to some extent; (3) During welding, the pin of FSW tool is inserted into the welding materials and surrounded by the material flows. This brings difficulty for *in-situ* tool condition monitoring (Scialpi *et al*., 2008). As a result, although tool wear in FSW is important, it is usually considered as a pre-determined condition for studies on other aspects, such as the evaluation of joint quality, the modeling for material flows, the computations of welding dynamics and so on (Kumar and Raju, 2012; Gibson *et al*., 2014b; Hirasawa *et al*., 2010). Tool condition in FSW is usually obtained in an indirect way by calculations via physics-based methods, e.g., the computational fluid dynamics methods on the governing equations about mass, momentum, and energy with boundary conditions (Hirasawa *et al*., 2010; Elangovan *et al*., 2008).

Although *in-situ* tool inspection is usually difficult to perform, there are many indirect inspection approaches to obtain the tool wear condition in FSW or μFSW, e.g., measuring an impression of the tool and then using the inverse of the coupon as a surrogate of the tool surface image. Since tool surface measurements obtained at different stages of tool wear contain both spatial and temporal dependences, we are inspired to develop a spatio-temporal model to characterize tool wear progression. We will review spatio-temporal modeling methods and the model fitting methods in the following sections.

2.4.2 Spatio-temporal Model

Two major strategies are usually adopted to build spatio-temporal models: the joint probability based Geostatistical modeling strategy (Kyriakidis and Journel, 1999) and the

conditional probability based hierarchical modeling strategy (Wikle *et al*., 1998). We focus on hierarchical modeling. Details about the Geo-statistical modeling strategy can be referred to Kyriakidis and Journel (1999).

Hierarchical spatio-temporal modeling is based on Bayesian inference. The model fitting results are the posterior distributions of the spatio-temporal process given observations and parameters (Wikle *et al*., 1998). Based on the review in Wikle *et al*. (1998) and the derivation in Berliner *et al*. (2000), a hierarchical spatio-temporal model decomposes a complete joint probability model that covers all data, process, and parameters into the product of a series of conditional models (Berliner *et al*., 2000). The decomposed conditional models, for different purposes, are arranged in five stages. The first stage is to model the measured spatio-temporal observations, focusing on the spatial measurement. In spatio-temporal statistics, the spatio-temporal observations are regarded as a sample collected from a spatio-temporal process (Wikle *et al*., 1998) and the observations are usually expressed as a linear model of the process and an error term. The second stage is to model the temporal and spatial features in different scales, e.g., the seasonal trend for atmospheric (Wikle *et al*., 1998) and the monthly behavior of oceanographic processes (Berliner *et al*., 2000). Some spatial structures are also modeled in the second stage (Katzfuss and Cressie, 2012; Cressie *et al*., 2010). The third stage is to model the spatial structure and the temporal dynamics. The fourth stage is to give the prior distributions to the parameters in the first three stages. The fifth stage is to model the hyperparameters of the priors in the fourth stage, if necessary.

The probability decomposition and the five-stage modeling provide great convenience and flexibility for using a hierarchical model to study and analyze a spatio-

temporal process in terms of goodness of fit, interpretability, parsimony, prediction capability, and computational cost (Cameletti *et al*., 2011). Hence, hierarchical models are widely adopted to study the complicated spatio-temporal processes such as climate (Cameletti *et al*., 2013), oceanography (Berliner *et al*., 2000; Wikle *et al*., 2001), air pollution (Cameletti *et al*., 2011), ecology (Cosandey-Godin *et al*., 2014), etc. These spatio-temporal processes all have the properties that the space-time data are usually in high-dimension with complicated multi-scale formats; the process evolves along complex dynamics and many parameters need to be estimated.

Among all the spatio-temporal processes that are suitable for hierarchical modeling, the general physics-based dynamic spatio-temporal process should be emphasized in this section. Different from the pure data-driven studies listed above, the physics-based dynamic process adopts scientific disciplines to describe spatially-explicit processes that evolve over time (Wikle and Hooten, 2010). These scientific disciplines are usually characterized by deterministic dynamic models, e.g., partial differential equations, integro-difference equations, matrix models, agent-based models and etc. The biggest challenge with such dynamic models is the curse of dimensionality. Hierarchical models have proven invaluable in their abilities to some extent to deal with this issue (Wikle and Hooten, 2010) and therefore, more and more science-based dynamic spatio-temporal processes have been modeled using a hierarchical structure (Berliner, 2003; Calder, 2008; Holton, 1973; Stroud *et al*., 2010). We need to point out that the research in Chapter 5 is a hybrid model to combine the traditional data-driven method with the physics-based dynamic process under the hierarchical structure. This will be a new development on tool wear monitoring.

2.4.3 Fitting Spatio-temporal Model

Besides the benefits in spatio-temporal modeling, the hierarchical model structure has advantages in computations for model fitting and parameter estimations. Three widely used approaches for fitting hierarchical spatio-temporal models are Markov Chain Monto Carlo (MCMC), Kalman filter, and a recently developed method called integrated nested Laplace approximation (INLA).

MCMC is the most widely used approach despite its less satisfactory efficiency. To improve the efficiency of MCMC, new methods have been developed in the MCMC literature: the "delayed-rejection adaptive Metropolis" (DRAM) sampling algorithm (Haario *et al*., 2006) for improving the convergence in Metropolis-Hastings steps for highly nonlinear models, the "multiple very fast simulated annealing" (MVFSA) algorithm in the context of parameter estimation in climate models (Villagran *et al*., 2008), and the further improvements in "differential evolution adaptive Metropolis" (DREAM) algorithm (Vrugt *et al*., 2009).

One alternative class of methods in the MCMC literature is the sequential Monte Carlo methods (i.e., particle or bootstrap filtering and ensemble Kalman filtering) (Wikle and Hooten, 2010). Approaches that are appealing in sequential Monte Carlo include independent simulation, importance sampling, sequential importance sampling, or particle filtering (Gordon *et al*., 1993; Liu, 2008). However, disadvantages of the sequential Monte Carlo methods have been pointed out. Most importance sampling methods and particle filtering can suffer from sample impoverishment. The sampling methods under sequential Monte Carlo have been criticized for not utilizing the current

data in the sampling step (Liu, 2008). Many verification methods have been developed to overcome these issues. Smith (2013) provides a detailed discussion on these methods.

Kalman filter, especially the ensemble Kalman filter can be regarded as a special sequential Monte Carlo method that relies on Gaussian assumptions to approximate the posterior distribution of the unobserved state variable. Stroud *et al*. (2010) provides a nice discussion on using Kalman filter to assimilate the satellite images for sediment transport modeling in Lake Michigan. Adopting Kalman filter to fit a hierarchical spatio-temporal model for process monitoring has a unique benefit that the outputs from Kalman filter can characterize the process information, based on which the process anomalies can be detected.

INLA is a numerically implemented analytical solution for approximating the marginal posterior distributions in a hierarchical model with latent Gaussian processes. The major advantage of INLA over other stochastic solutions (MCMC, importance sampling) is that INLA can find the marginal posterior distribution faster and with more accurate estimation results on the variances. Details of INLA can be referred to Rue *et al*. (2009) and Rue and Held (2005). Despite INLA's satisfactory efficiency, it suffers from applicability as a wide range of models cannot be fitted via INLA, including the science-based dynamic function with a transition matrix.

Considering the advantages and disadvantages of the available model fitting methods, we select Kalman filter to fit the hierarchical spatio-temporal model for μFSW tool wear progression. The proposed method will be presented in Chapter 5.

2.5 Limitations of Literature

Based on a thorough review of the related literature, the limitations of current works and research gaps are summarized as follows.

1. There is a need for *in-situ* non-destructive evaluation of advanced joining techniques, especially in joining dissimilar lightweight materials. Although destructive testing and offline NDEs have been extensively investigated, they can only provide quality information after the process, at which point quality defects have already occurred and can no longer be remedied. Existing NDEs also heavily rely on the expertise and experience of the technician. Sensor signals captured during the joining process provide rich information that can be used to monitor the process in real-time and provide quality prediction for the product. Real-time quality evaluation can potentially trigger process controls to fix a potential defect. By the time the process finishes, corrective actions have already been taken and the output product is in high quality. Therefore, *in-situ* non-destructive methods that provide near real-time evaluation of product quality need to be developed.
2. Research on monitoring the multi-profile data from FSBR process is extremely limited. The only related work is Guo *et al*. (2017). However, Guo *et al*. (2017) treated the force and torque signals separately, ignoring the correlation between the signals; process settings, which are critical information for NDE, were not considered either; further, given the small training sample size, the decision-making method in Guo *et al*. (2017) didn't consider the possible high variance or high bias of the evaluation results. Therefore, sensor fusion and process

monitoring methods that consider sensor-to-sensor correlation, process settings, and the small sample size scenario need to be developed.

3. There is no research on the integration of tensor decomposition and ensemble learning. Although tensor decomposition and ensemble learning are effective in their respective areas, no research on how to best integrate them has been launched yet. This integration should be strategically done instead of a naïve integration. On one hand, limited training sample size, which is common in many manufacturing processes, can jeopardize the accuracy in ensemble learning. On the other hand, tensor decomposition methods may be unstable and the resulting features are sensitive to certain parameters in the decomposition; this requires the subsequent ensemble learning method to be able to stabilize the feature extraction. Therefore, an optimal integration of tensor decomposition and ensemble learning is needed.
4. Tool wear is a complicated spatio-temporal process. While an accurate physics-based model for μFSW tool wear is not yet available, there is a huge interest in a data-driven spatio-temporal model to characterize tool wear progression. A hybrid hierarchical model that combines physics and regression is promising to effectively model tool wear progression and enable tool condition monitoring.

In summary, new methods are needed in *in-situ* non-destructive evaluation of advanced joining techniques, sensor fusion and process monitoring with sensor-to-sensor correlation and small sample size, optimal integration of tensor decomposition and ensemble learning, and spatio-temporal modeling and monitoring of tool wear progression.

CHAPTER 3 SENSOR FUSION AND ON-LINE MONITORING OF FRICTION STIR BLIND RIVETING FOR *IN-SITU* NDE IN LIGHTWEIGHT MATERIALS MANUFACTURING

In this chapter, we develop a sensor fusion and online process monitoring method to obtain a better understanding of FSBR to enable *in-situ* NDE. The method is developed for monitoring the joining of CFRP composites and aluminum alloy 6111 (AA6111) sheets, and it can be easily extended to other materials. Experiments have shown that the quality of a FSBR joint is affected by many factors, including process parameters, material stack-up sequence, quality of the sheets, fixture setup, and tool condition. Preliminary analysis has shown that most of the impact from these factors can be reflected in the penetration force and torque signals, which are collected in real-time during the FSBR process. This study focuses on establishing the relationship between these *in-situ* signals and the strength of the joint. In the proposed research, features extracted from the *in-situ* signals are combined with engineering knowledge to provide quality evaluations for the joints. This chapter is organized in Gao *et al*. (2018) and received the best paper award in ASME 2018 conference.

In this research, the penetration force and torque data are recorded at discrete time points with equal time intervals. The signals have complicated patterns and the measurement may contain noise, irrelevant or redundant information. In order to effectively analyze such multi-sensor nonlinear profile data, we propose to represent the profiles in high-dimensional arrays and use tensor decomposition methods to extract features from the original multi-sensor data.

In the proposed method, two unsupervised tensor decomposition algorithms – multilinear principal component analysis (MPCA) and uncorrelated multilinear principal component analysis (UMPCA) are introduced to extract features from the tensorial representation of FSBR force and torque signals. The extracted features are then combined with engineering-driven features and experimental settings to provide information about the FSBR process. While most features are continuous, the experimental settings are represented as categorical features. To select the most significant features from the mixed-type data, we adopt the sparse group lasso regression method to select features with sparsity consideration. Selected features are then fed into classifiers and the classification results indicate the predicted quality of FSBR joints. Due to limited sample size, the results from an individual classifier may have either a high training bias or a high training variance and thus providing poor classification accuracy. We propose to use classifier fusion to provide a more accurate result by balancing the biases and variances from different classifiers. Five individual classifiers are fused; their weights are obtained by optimizing the classification accuracy. By integrating feature extraction, feature selection, and weighted classifier fusion, the proposed method provides online monitoring of the FSBR process and accurate prediction of the joint quality in real-time.

3.1 Data Description

In this study, the FSBR process fabricated lap sheer joints from two material stack-combination, AA6111-CFRP (AA6111 is the top sheet) and CFRP-AA6111 (CFRP is the top sheet), at various spindle speeds (ω = 3,000 and 5,000 rpm) and feed rates (ν = 120 and 420 mm/min). The strengths of selected joints were evaluated in tensile test. Table

3.1 lists the number of samples produced in each experiment setting; the number in parenthesis represents the number of samples with tensile test result. Not all joints were tested due to time and cost constraints. Among the total $N = 74$ joints produced, $N_1 = 18$ joints had tensile test result and $N_2 = 56$ joints were not tested.

Table 3.1 Number of FSBR samples

Feed rate ν (mm/min)	Spindle speed ω (rpm)	Number of samples (# with tensile test)	
		AA6111-CFRP	CFRP-AA6111
120	3000	4 (1)	3 (1)
	5000	4 (3)	49 (8)
420	3000	3 (0)	3 (2)
	5000	4 (0)	4 (3)

3.1.1 Force and Torque Signals

During the process, a force signal and a torque signal were captured for each joint. A pair of force and torque values were recorded every 0.01 seconds. Figure 3.1(a) shows the raw signals in fabricating an AA6111-CFRP joint under $\omega = 3{,}000$ rpm and $\nu = 120$ mm/min. It is noted that the raw dataset contains a lot of irrelevant data points recorded before and after the actual FSBR process. To remove the irrelevant data points in the raw signals, we identified the start and end time of each process and then extracted 6 seconds of each signal to account for the duration of the process.

We also notice from Figure 3.1(a) that the signals contain a lot of noise. To de-noise the signals, we used wavelet method to localize the patterns in the signal to different scales, thus preserving important signal features while removing noise. Figure 3.1(b) shows the de-noised and truncated signals for the raw signals in Figure 3.1(a). It

can be seen that wavelets can effectively de-noise the force and torque signals while preserving their original complex shapes.

After pre-processing, the force and torque signals are represented in a 3-way array $\mathcal{A}$, which is a tensor object $\mathcal{A} \in \mathbb{R}^{600\times2\times N}$, where 600 is the number of data points collected on each signal, 2 is the number of sensors, and $N = 74$ is the number of samples. Figure 3.1(b) shows that the pre-processed signals are nonlinear with complex shapes and contain sensor-to-sensor correlation. Therefore, tensor decomposition methods are needed to extract features from $\mathcal{A}$.

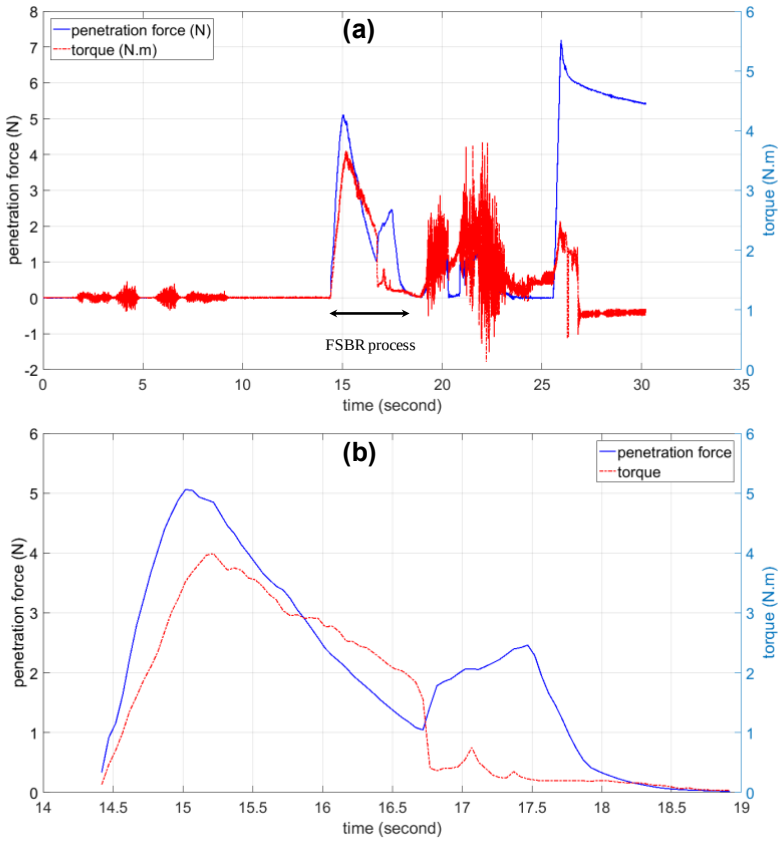


Figure 3.1 Force and torque signals of an AA6111-CFRP joint under ω = 3,000 rpm and v = 120 mm/min: (a) raw signals and (b) pre-processed signals.

3.1.2 Tensile Test and Joint Quality

In tensile test, the maximum tensile load is the load under which the FSBR joint starts to fracture. The maximum tensile loads of the joints that had tensile tests were extracted and used as the quality response of those joints. These joints were then clustered into two quality groups based on their maximum tensile loads. Figure 3.2 shows the dendrogram of hierarchical clustering.

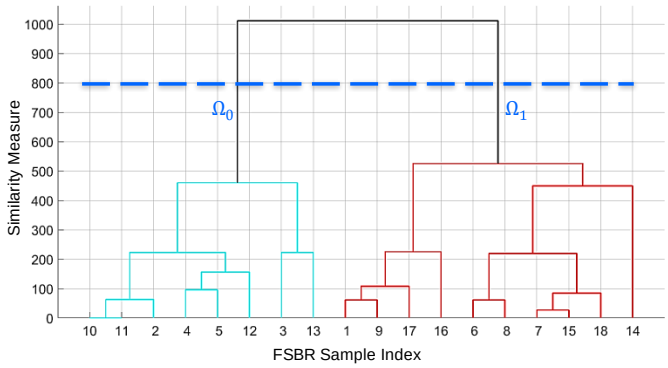


Figure 3.2 Clustering result of the FSBR samples: low-quality group $\boldsymbol{\Omega_0 = \{2, 3, 4, 5, 10, 11, 12, 13\}}$ and high-quality group $\boldsymbol{\Omega_1 = \{1, 6, 7, 8, 9, 14, 15, 16, 17, 18\}}$.

Based on hierarchical clustering, two quality groups are obtained: low-quality group, denoted by $\Omega_0 = \{i: L_i < L_0, i = 1, \cdots, N_1\}$, and high-quality group, denoted by $\Omega_1 = \{i: L_i > L_0, i = 1, \cdots, N_1\}$, where L_i is the maximum tensile load of sample i and L_0 is the decision boundary. Based on the quality groups, we assign the quality response of sample i to be $y_i = 0$ if $i \in \Omega_0$ and $y_i = 1$ if $i \in \Omega_1$. As shown in Figure 3.2, among the $N_1 = 18$ joints that had tensile tests, 8 samples are in the low-quality group, $\Omega_0 = \{2, 3, 4, 5, 10, 11, 12, 13\}$, and 10 samples are in the high-quality group, $\Omega_1 = \{1, 6, 7, 8, 9, 14, 15, 16, 17, 18\}$.

3.2 Method Overview

The key steps in the proposed method for FSBR process monitoring and quality prediction are shown in Figure 3.3. Information about the FSBR process includes parameters in the experiment setup, material stack-up sequence, and the recorded force and torque signals. In Step 1, engineering-driven features are extracted from the signals directly based on our understanding of the FSBR process. Two unsupervised tensor decomposition algorithms are introduced to extract features from the 3-way array $\mathcal{A}$, based on multilinear tensor-to-vector or tensor-to-tensor projections. In Step 2, sparse group lasso regression is applied to select the most significant features from the extracted features. Selected features are then fed into five individual classifiers in Step 3. The results from individual classifiers are fused with optimal weights to obtain the final classification results that indicate the predicted quality of FSBR joints. Quality information of the FSBR joints are needed in Step 2 and Step 3 so that the best features can be selected and that the classifiers are optimized. The details of each step will be elaborated in Sections 3.3, 3.4, and 3.5.

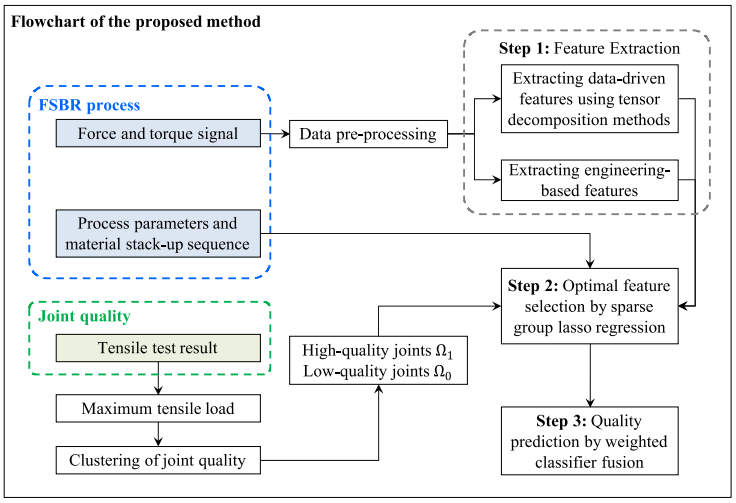


Figure 3.3 Flowchart of the proposed method

By integrating feature extraction, feature selection, and classifier fusion, the proposed method establishes the relationship between FSBR process and joint quality. The proposed method is then applied to the samples that didn't have tensile test results so that their quality can be estimated. The proposed method will also provide online monitoring of the FSBR process and accurate *in-situ* NDE of the joint quality.

3.3 Feature Extraction

It is critical to effectively extract features from the raw signals to provide useful information about the FSBR process. Experimental settings – feed rate, spindle speed, and material stack-up sequence – are represented as categorical features. Each feature has two levels, as listed in Table 3.2. Section 3.3.1 will elaborate the details on extracting engineering-driven features based on our understanding of the process. Section 3.3.2 will introduce unsupervised tensor decomposition algorithms for extracting data-driven features.

Table 3.2 Summary of categorical features

Feature	Description
AC/CA	AA6111-CFRP (AA6111 is the top sheet) or CFRP-AA6111 (CFRP is the top sheet)
Feed rate	ν = 120 or 420 mm/min
Spindle speed	ω = 3000 or 5000 rpm

3.3.1 Extracting Engineering-Driven Features

A total of 12 features are defined based on expert knowledge of the FSBR process, such as the maximum force value during the penetration of the top sheet and the duration of the process. These features are illustrated in Figure 3.4 and described in Table 3.3. All these features are continuous.

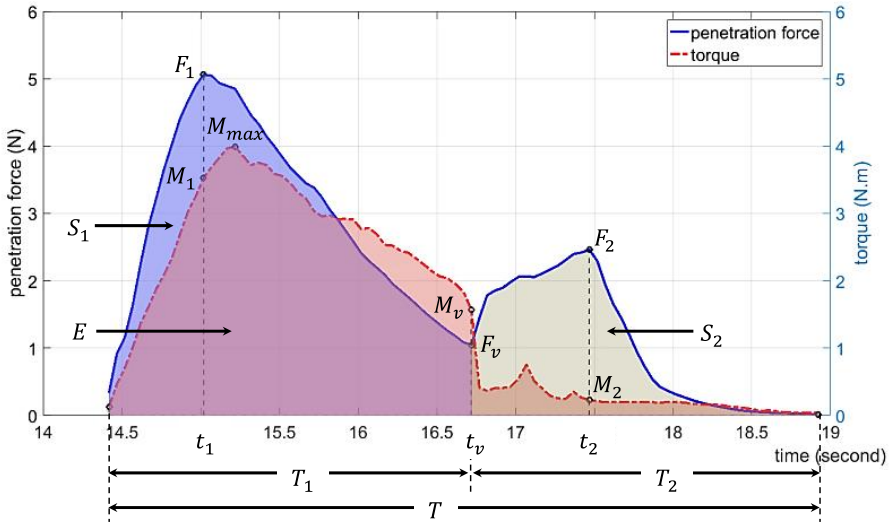


Figure 3.4 Engineering-driven features from Figure 3.1's AA6111-CFRP joint.

During the penetration of the top workpiece in FSBR, the maximum force F_1 is recorded at time t_1, at which time the torque is M_1. At time t_v, the rivet penetrates through the top workpiece and touches the interface between the top and bottom workpieces; the force at time t_v is F_v and the torque is M_v. The rivet then continues to penetrate the bottom workpiece and the maximum force F_2 is recorded at time t_2 with torque M_2. In addition, the maximum torque during FSBR is denoted as M_{max}. The duration of the entire FSBR process is T and the duration of the penetration of the top sheet is T_1. Features S_1 and S_2 represent the energy consumed by the force during the penetration of the top sheet and bottom sheet, respectively. Similarly, feature E represents the energy consumed by the

torque during FSBR. The energy features are calculated as the area under the force or torque curve.

Table 3.3 Summary of engineering-driven features

Feature	Description
F_1	Maximum force value during the penetration of the top sheet
F_2	Maximum force value during the penetration of the bottom sheet
F_v	Valley force value during the penetration towards the bottom sheet
M_1	Torque value when F_1 is recorded
M_2	Torque value when F_2 is recorded
M_v	Torque value when F_v is recorded
M_{max}	Maximum torque value during the FSBR process
T_1	Time duration of the penetration of the top sheet
T	Time duration of the entire FSBR process
S_1	Area under the force signal in T_1
S_2	Area under the force signal in $T_2 = T - T_1$
E	Area under the torque signal in T

3.3.2 Extracting Data-Driven Features

Although engineering-driven features can effectively represent certain characteristics about the process, a lot of important information is still hidden in the complex shapes of the signals and in the sensor-to-sensor correlations. Therefore, feature extraction methods are needed to effectively reduce the dimensionality of the original signals and to extract data-driven features that contain important information about the FSBR process.

As reviewed in Chapter 2, multi-linear extensions of PCA allow applying the PCA technique to tensors without unfolding the original dataset. The advantages of the multi-linear approach over their vectorized versions include the higher efficiency in terms

of computational costs and memory demands, the easier interpretation of retained PCs, and the possibility to better characterize the actual multi-linear correlation structure. Two unsupervised tensor decomposition algorithms, multilinear principal component analysis (MPCA) (Lu *et al*., 2008a) and uncorrelated multilinear principal component analysis (UMPCA) (Lu *et al*., 2008b), for multi-sensor data are described hereafter. The tensor representation of the FSBR force and torque signals is $\mathcal{A} \in \mathbb{R}^{L \times J \times N}$, where $L = 600$ is the number of data points collected on each signal, $J = 2$ is the number of sensors, and N is the number of samples.

3.3.2.1 Basic Multilinear Algebra

Without loss of generality, the object $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is a tensor of order M = 3 and it is addressed by 3 indices $i_m \in \{1, 2, \dots, I_m\}, m = 1,2,3$. I_m represents the dimension of the m-mode of $\mathcal{A}$, in which an m-mode vector is defined as a vector obtained by varying i_m, while keeping other indices fixed (Kolda and Bader, 2009; Lu *et al*., 2008a). The m-mode unfold of a tensor $\mathcal{A}$ is a matrix represented by $\boldsymbol{A}_{(m)}$, whose columns are the m-mode vectors of $\mathcal{A}$, i.e., $\boldsymbol{A}_{(1)} \in \mathbb{R}^{I_1 \times (I_2 I_3)}$, $\boldsymbol{A}_{(2)} \in \mathbb{R}^{I_2 \times (I_1 I_3)}$, and $\boldsymbol{A}_{(3)} \in \mathbb{R}^{I_3 \times (I_1 I_2)}$.

A tensor can be multiplied by a matrix along a mode. The multiplication between a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ and a matrix $\boldsymbol{B} \in \mathbb{R}^{J_m \times I_m}$ along the m^{th} mode of $\mathcal{A}$ is denoted as $\mathcal{A} \times_m \boldsymbol{B}$. The multiplication result is an $I_1 \times \cdots \times I_{m-1} \times J_m \times I_{m+1} \times \cdots I_M$ tensor. The scalar product of two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is $\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1} \sum_{i_2} \sum_{i_3} \mathcal{A}(i_1, i_2, i_3)\, \mathcal{B}(i_1, i_2, i_3)$ and the Frobenius norm of tensor $\mathcal{A}$ is calculated as $\|\mathcal{A}\|_F = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$.

3.3.2.2 The MPCA Approach

The basic MPCA approach is a direct extension of the regular PCA technique to higher dimensions (Friedman *et al*., 2001; Lu *et al*., 2008a). Since the third mode in $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ represents the number of samples, without loss of generality, we let $\mathcal{A}_n$ denote the n^{th} sample of the two-dimensional tensor object, $\mathcal{A}_n \in \mathbb{R}^{I_1 \times I_2}$.

The objective of MPCA is to find a set of orthogonal transformation matrices that projects the original tensor $\mathcal{A}_n \in \mathbb{R}^{I_1 \times I_2}$ into a low-dimensional tensor $\mathcal{S}_n \in \mathbb{R}^{P_1 \times P_2}$:

$$\mathcal{S}_n = \mathcal{A}_n \times_1 \boldsymbol{H}^{(1)T} \times_2 \boldsymbol{H}^{(2)T}, \tag{3.1}$$

where $\boldsymbol{H}^{(m)} \in \mathbb{R}^{I_m \times P_m}$ are the transformation matrices; P_1 and P_2 are the projected dimensions with $P_m \leq I_m$, $m = 1, 2$. The projected low-dimensional tensor should capture the most of variation observed in the original tensor:

$$\left\{\boldsymbol{H}^{(1)}, \boldsymbol{H}^{(2)}\right\} = \arg\max_{\boldsymbol{H}^{(1)}, \boldsymbol{H}^{(2)}} \sum_{n=1}^{I_3} \|\mathcal{S}_n\|_F^2 , \tag{3.2}$$

where $\|\cdot\|_F$ is the Frobenius norm. More details on the MPCA model can be found in Lu *et al.* (2008a).

The projected dimensions P_1 and P_2 can be determined by setting a threshold θ on the percentage of the total variation that is to be kept in $\mathcal{S}$, $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_{I_3})$ and $\mathcal{S} \in \mathbb{R}^{P_1 \times P_2 \times I_3}$. The residuals are obtained by $\mathcal{E} = (\mathcal{E}_1, \ldots, \mathcal{E}_{I_3})$ and

$$\mathcal{E}_n = \mathcal{A}_n - \mathcal{S}_n \times_1 \boldsymbol{H}^{(1)T} \times_2 \boldsymbol{H}^{(2)T} . \tag{3.3}$$

$\mathcal{S}_n$ and $\mathcal{E}_n$ are then vectorized to form the extracted features. Therefore, a total of $(P_1 P_2 + 1)$ features are extracted by MPCA, which include $(P_1 P_2)$ principal components

and one residual feature. It should be noted that the extracted features by MPCA are correlated sometime.

3.3.2.3 The UMPCA Approach

To overcome the problem of correlated features in MPCA, UMPCA was developed in Lu *et al.* (2008b). The UMPCA approach follows the classical PCA derivation of successive variance maximization to enforce the uncorrelated features. Different from the tensor-to-tensor projection in MPCA, UMPCA applies tensor-to-vector projection to map the original tensor into a vector via a number of elementary multilinear projections (EMPs).

The projection of the n^{th} sample $\mathcal{A}_n$ into the p^{th} subspace $\left\{\boldsymbol{u}_p^{(1)T}, \boldsymbol{u}_p^{(2)T}\right\}$ is denoted as:

$$s_n(p) = \mathcal{A}_n \times_1 \boldsymbol{u}_p^{(1)T} \times_2 \boldsymbol{u}_p^{(2)T}, \tag{3.4}$$

where $s_n(p)$ is the p^{th} PC-score of the n^{th} sample, $p = 1, \dots, P_0$. The objective of UMPCA is to find the EMPs $\left\{\boldsymbol{u}_p^{(1)T}, \boldsymbol{u}_p^{(2)T}\right\}_{p=1}^{P_0}$ that maximize the captured variance:

$$\left\{\boldsymbol{u}_p^{(1)T}, \boldsymbol{u}_p^{(2)T}\right\} = \arg\max_{\boldsymbol{u}_p^{(m)}} \sum_{n=1}^{I_3} \left(s_n(p) - \bar{s}_p\right)^2, \tag{3.5}$$

$$\text{subject to} \quad \boldsymbol{u}_p^{(m)T}\boldsymbol{u}_p^{(m)} = 1, m = 1,2$$

$$\frac{\boldsymbol{g}_p{}^T \boldsymbol{g}_q}{\|\boldsymbol{g}_p\|\|\boldsymbol{g}_q\|} = \delta_{pq}, p, q = 1, \dots, P_0.$$

$\bar{s}_p$ is the average of the p^{th} PC-scores. $\sum_{n=1}^{I_3}\left(s_n(p) - \bar{s}_p\right)^2$ is the variance of the projected vectors; $\delta_{pq} = 1$ if $p = q$ and 0 if $p \neq q$. The $\frac{\boldsymbol{g}_p{}^T \boldsymbol{g}_q}{\|\boldsymbol{g}_p\|\|\boldsymbol{g}_q\|} = \delta_{pq}$ constraint guarantees that the projected features are uncorrelated. More details on UMPCA can be found in Lu *et al.* (2008b).

In UMPCA, the number of PC-features is limited by $P_0 \leq \min\{\min\{I_m\}, M\}$, where M is the order of input tensor $\mathcal{A}$. Since there are two signals in this study, $I_2 = 2$, we have $P_0 \leq 2$. The number of features extracted by UMPCA is limited to two.

3.4 Optimal Feature Selection

The FSBR process is adequately represented by the extracted features, which include categorical features that represent the experimental settings, 12 continuous features from the *in-situ* force and torque signals based on engineering knowledge, and $(P_1P_2 + 1)$ continuous features from MPCA or $P_0 \leq 2$ continuous features from UMPCA. Let $\mathcal{F}$ denote the set of extracted features and $P = |\mathcal{F}|$ is the total number of extracted features.

Although a total of N = 74 joints was fabricated, only the N_1 = 18 joints that had tensile test can be used to train the quality evaluation model. Since N_1 is not large comparing to P, feature selection is needed to identify the most informative features. Feature selection allows the simplification of the quality prediction model so that it's easier to interpret. A good selection of features also enables shorter model training time, helps to avoid the curse of dimensionality (especially if P_1P_2 is relatively large), and enhances model generalization by reducing overfitting (Bermingham *et al*., 2015).

Since both continuous and categorical features are present in the extracted features, the feature selection method in this study should effectively handle mixed-type data. Therefore, we propose to adopt the sparse group lasso (SGL) regression method as a feature selection method in this study. Categorical features are treated as grouped features in SGL. Each individual continuous feature is treated as a group of size one. For the purpose of feature selection, SGL is superior to group lasso or regular lasso in the sense

that SGL selects features with the consideration of groupwise and within-group sparsity (Friedman *et al*., 2001; Yuan and Lin, 2006; Simon *et al*., 2013).

Let $\boldsymbol{X}$ denote the $N \times P$ matrix of extracted features for all samples and $\boldsymbol{X}_1 \in \mathbb{R}^{N_1 \times P}$ denote the matrix of extracted features for the N_1 samples with quality response. Let $\boldsymbol{y}$ denote the response vector of length N_1. The SGL regression extends lasso regression by introducing two penalty terms to the objective function:

$$\widehat{\boldsymbol{\beta}}_{SGL} = \text{argmin}_{\boldsymbol{\beta}} \frac{1}{2N_1} \left\| \boldsymbol{y} - \sum_{k=1}^{K} \boldsymbol{X}_1^{(k)} \boldsymbol{\beta}^{(k)} \right\|_2^2 + (1-\alpha)\lambda \sum_{k=1}^{K} \left\| \boldsymbol{\beta}^{(k)} \right\|_2 + \alpha\lambda \|\boldsymbol{\beta}\|_1, \tag{3.6}$$

where K is the number of feature groups, $\boldsymbol{X}_1^{(k)}$ is the submatrix of $\boldsymbol{X}_1$ with columns corresponding to the predictors in the k^{th} feature group, $k = 1, \cdots, K$. The k^{th} group has J_k features and thus $\boldsymbol{X}_1^{(k)} \in \mathbb{R}^{N_1 \times J_k}$. Vector $\boldsymbol{\beta}^{(k)} \in \mathbb{R}^{J_k \times 1}$ is the coefficient vector of the k^{th} group. $\boldsymbol{\beta} = \{\boldsymbol{\beta}^{(k)}\}_k$ represents the regression coefficients for all groups.

In Eq. (3.6), $\left\| \boldsymbol{y} - \sum_{k=1}^{K} \boldsymbol{X}_1^{(k)} \boldsymbol{\beta}^{(k)} \right\|_2^2$ is the sum of squared errors for the grouped features; $\sum_{k=1}^{K} \left\| \boldsymbol{\beta}^{(k)} \right\|_2$ is the l_2 norm to account for the number of groups selected; $\|\boldsymbol{\beta}\|_1$ is the l_1 norm to account for the number of individual features selected. The two penalty terms are controlled by the groupwise sparsity parameter α and within-group sparsity parameter λ, respectively. If $\alpha = 1$, Eq. (3.6) reduces to lasso regression; if $\alpha = 0$, SGL reduces to group lasso regression. The SGL model can be fitted via an accelerated generalized gradient algorithm with backtracking (Friedman *et al*., 2001; Yuan and Lin, 2006; Simon *et al*., 2013).

In this study, features are selected based on our knowledge of $\boldsymbol{X}_1$ and $\boldsymbol{y}$. The optimal values for α and λ are determined by leave-one-out cross-validation. The

selected features are represented in set $\tilde{\mathcal{F}} \subseteq \mathcal{F}$ with $\tilde{P} = \left|\tilde{\mathcal{F}}\right| \leq P$. $\widetilde{\boldsymbol{X}}_1 \in \mathbb{R}^{N_1 \times \tilde{P}}$ is the matrix of selected features for the N_1 samples with quality response. Applying $\tilde{\mathcal{F}}$ to $\boldsymbol{X}$, we obtain $\widetilde{\boldsymbol{X}} \in \mathbb{R}^{N \times \tilde{P}}$ as the matrix of selected features for all the N samples; let $\widetilde{\boldsymbol{X}}_2 \in \mathbb{R}^{N_2 \times \tilde{P}}$ denote the matrix of selected features for the N_2 samples without quality response.

3.5 Quality Evaluation

In this section, we develop a weighted classifier fusion method to establish the relationship between the selected features and the quality response of the FSBR joints. Since the training dataset obtained from FSBR experiments is usually small, the classification result from an individual classifier tends to have either a large bias or a large variance. Thus, arbitrarily choosing an individual classifier for FSBR quality evaluation may lead to poor results. A remedy for overcoming this drawback is to integrate multiple classifiers.

Classification fusion combines the classification results of individual classifiers to conclude a final classification result. Fusing classifiers' decisions can improve the classification accuracy in the decision-making process (Ruta and Gabrys, 2000). In the traditional weighted majority voting approach, the outputs of individual classifiers are weighted and linearly summed up and the class label with the largest weight is chosen as the final classification fusion result (Littlestone and Warmuth, 1994). Instead of fusing the class labels, we propose a fusion approach that weights the probability outputs of individual classifiers and then sums them up to estimate a final probability, based on which the class label will be assigned.

Since quality evaluation for FSBR is essentially a binary classification problem, the probability of assigning sample i to the high-quality group Ω_1 will be estimated in

each classifier. Let π_i^c denote the probability that classifier c assigns sample i to Ω_1. The probabilities π_i^c will be weighted and fused to obtain the final probability estimation. The proposed fusion approach is different from the traditional weighted majority voting approach since it relies more on the probability outputs rather than the class labels. The rationale behind the proposed approach is elaborated in the following intuitive example. If sample i has $\pi_i^1 = 0.95$ with class label 1 and $\pi_i^2 = 0.45$ with class label 0, assuming equal weights for the two classifiers, the traditional voting approach would produce a tie between label 0 and label 1, whereas the proposed approach would give a final probability of $\pi_i = (\pi_i^1 + \pi_i^2)/2 = 0.7$, which indicates that class label 1 should be assigned to sample i. The result from the proposed fusion approach is consistent with our intuition since π_i^1 is close to 1, indicating high confidence in the decision, while π_i^2 is close to 0.5, indicating uncertainty in the classification.

In the proposed weighted classifier fusion approach, $\widetilde{\boldsymbol{X}}_1$, the matrix of selected features for the N_1 samples with quality response, and the quality response $\boldsymbol{y}$ are fed into five individual classifiers to obtain the probability outputs, which are then fused to obtain the FSBR quality evaluation model.

3.5.1 Individual Classifiers

The individual classifiers adopted in the proposed fusion approach are logistic regression, kernel support vector machine (KSVM) with polynomial kernel, KSVM with gaussian kernel, neural network, and k-nearest neighbors. These five classifiers are chosen in order to address different types of patterns in the dataset.

Logistic regression models the posterior probabilities of binary classes in a linear model and thus is most suitable if the classes can be linearly separated. KSVM is able to

handle more complicated patterns in binary classification. The polynomial kernel is an extension of the logistic regression for nonlinear decision boundaries and the gaussian kernel is a more generalized version that assigns weights to points based on the Gaussian density function. Neural network is a two-stage classification model that generates nonlinear functions of the linear combinations among the predictors. Using multiple hidden layers in a neural network allows the construction of hierarchical features at different levels of resolution, thus providing good flexibility in capturing the nonlinearities in the data. The back-propagation network is used in this research. k-nearest-neighbor (KNN) classifier is a model-free classification method that adopts a voting mechanism to integrate the results from the neighborhood of each query point. Details about the classifiers can be found in Friedman *et al*. (2001). In this study, the parameters in each individual classifier, such as the penalty cost parameter in KSVM and the number of hidden layers in neural network, are determined by cross-validation.

3.5.2 Weighted Classifier Fusion

The estimated probability that classifier c assigns sample i to Ω_1 is denoted as $\hat{\pi}_i^c$. The probabilities from individual classifiers are weighted and linearly summed up to obtain the fused probability estimation:

$$\hat{\pi}_i = \sum_{c=1}^{C} w_c \times \hat{\pi}_i^c \,, \tag{3.7}$$

where w_c is the weight for the c^{th} classifier; $\sum_{c=1}^{C} w_c = 1$ and $0 \le w_c \le 1, c = 1, \cdots, C$. $\hat{\pi}_i$ is the final estimated probability that sample i is assigned to Ω_1. The class label is then assigned by comparing $\hat{\pi}_i$ to a threshold π_0, represented in a logistic function as follows:

$$\hat{y}_i = \frac{1}{1 + e^{-a(\hat{\pi}_i - \pi_0)}} \,, \tag{3.8}$$

where $\hat{y}_i$ is the predicted class label for sample i; a is the convergence rate so that $\hat{y}_i = 1$ if $\hat{\pi}_i \geq \pi_0$ and $\hat{y}_i = 0$ if $\hat{\pi}_i < \pi_0$.

The performance of the weighted classifier fusion approach is evaluated by the correct classification rate (CCR). If all the N_1 samples are considered in this evaluation, then the CCR is estimated by:

$$\widehat{CCR} = 1 - \frac{|\hat{\boldsymbol{y}} - \boldsymbol{y}|^0}{N_1}, \tag{3.9}$$

where $\hat{\boldsymbol{y}}$ denotes the estimated class label vector of length N_1, calculated by Eq. (3.8); $\boldsymbol{y}$ is the true response vector as obtained by quality definition in Section 3.1. $|\hat{\boldsymbol{y}} - \boldsymbol{y}|^0$ is the l_0 "norm" that counts the number of non-zero entries of the vector $\hat{\boldsymbol{y}} - \boldsymbol{y}$.

In the proposed fusion approach, the weights w_c should be optimized so that the CCR can be maximized. We propose to obtain the optimal weights by cross-validation. In *k*-fold cross-validation, as each of the *k* subsamples is used as the validation set, we obtain a CCR result. The *k* CCR results from the folds can then be averaged to produce an average CCR, denoted as $\overline{CCR}$. The standard deviation of the *k* CCR results is s_{CCR}. Therefore, the optimal weights should maximize $\overline{CCR}$ while keeping s_{CCR} small. We adopt the definition of signal-to-noise ratio (SNR) and then obtain the optimal weights by:

$$\boldsymbol{w}^* = \arg\max_{\boldsymbol{w}} SNR = \arg\max_{\boldsymbol{w}} \overline{CCR}/s_{CCR}, \tag{3.10}$$

where $\boldsymbol{w}$ is the vector of w_c's.

Once the parameters, including the optimal weights, in the classifier fusion approach are obtained, the quality estimation model can be expressed as:

$$\hat{y}_i = \frac{1}{1 + e^{-a\left(\sum_{c=1}^{C} w_c^* \times \hat{\pi}_i^c - \pi_0\right)}}. \tag{3.11}$$

FSBR joint i is classified into the low-quality group Ω_0 if $\hat{y}_i = 0$; we assign $i \in \Omega_1$, the high-quality group, if $\hat{y}_i = 1$.

3.6 Application in Friction Stir Blind Riveting

In this section, the proposed method is demonstrated with the FSBR experiment data described in Section 3.1.

Data-driven features are extracted from tensor $\mathcal{A} \in \mathbb{R}^{600\times2\times74}$ of the FSBR force and torque signals by MPCA and UMPCA. In MPCA, by keeping θ = 80% of the total variation, $\mathcal{A}$ is projected onto a low-dimensional tensor $\mathcal{S} \in \mathbb{R}^{3\times1\times74}$. The L = 600 data points collected on each signal are projected onto $P_1 = 3$ features; the J = 2 sensors are projected onto $P_2 = 1$ feature. The residual tensor $\mathcal{E}$ then represents 20% of the total variation. A total of $(P_1P_2 + 1)$ = 4 features are extracted by MPCA. In UMPCA, since the number of extracted features is limited by 2, we extract P_0 = 2 uncorrelated features from $\mathcal{A}$. The data-driven features from either MPCA or UMPCA are then combined with the 12 engineering-driven features and 3 experimental setting parameters to form the feature set $\mathcal{F}$. $P = |\mathcal{F}| = 15 + (P_1P_2 + 1) = 19$ if MPCA is adopted and $P = 15 + P_0 = 17$ if UMCPA is adopted.

3.6.1 Feature Selection Results

In feature selection, we treat categorical features as grouped features in sparse group lasso. The feed rate and spindle speed are combined into a feature group of size four that represents the process parameters. “AC/CA” is treated as a group of size two that represents the material stack-up sequence. Each continuous feature is treated as a group of size one. Table 3.4 shows the results of optimal feature selection by sparse group lasso.

The optimal sparsity parameters are (α^*, λ^*), determined by leave-one-out cross-validation. When data-driven features from MPCA are adopted, the selected features are {F_1, PC_1, PC_3, AC/CA}, denoted by set $\tilde{\mathcal{F}}_A$, where PC_1 and PC_3 represent the first and third principal components extracted from MPCA. When data-driven features from UMPCA are adopted, the selected features are {F_1, PC_1, AC/CA}, denoted by set $\tilde{\mathcal{F}}_B$, where PC_1 represents the first principal component extracted from UMPCA. As listed in Table 3.3, F_1 is the engineering-driven feature that represents the maximum force during the penetration of the top sheet in FSBR.

We notice that the process parameters – feed rate v and spindle speed ω – are not selected by the (α^*, λ^*) SGL. Based on our understanding of FSBR, we prefer to include these features in the selected set. Therefore, in the $(\alpha = 0.95, \lambda^*)$ SGL, we fix α to 0.95 and then determine λ^* by leave-one-out cross-validation. When features from MPCA are adopted, the selected features are {F_1, PC_1, PC_3, AC/CA, v, ω}, denoted by set $\tilde{\mathcal{F}}_C$; when features from UMPCA are adopted, the selected features are {F_1, PC_1, AC/CA, v, ω}, denoted by set $\tilde{\mathcal{F}}_D$. We denote these four scenarios as A, B, C, and D, and we will compare their performance in classification.

Following the notations in Section 3.4, we let $\widetilde{\boldsymbol{X}}_A \in \mathbb{R}^{N \times \tilde{P}_A}$ denote the matrix of selected features of set $\tilde{\mathcal{F}}_A$ for all the N samples, $\tilde{P}_A = |\tilde{\mathcal{F}}_A|$. $\widetilde{\boldsymbol{X}}_A$ is further split into $\widetilde{\boldsymbol{X}}_{1A} \in \mathbb{R}^{N_1 \times \tilde{P}_A}$ and $\widetilde{\boldsymbol{X}}_{2A} \in \mathbb{R}^{N_2 \times \tilde{P}_A}$ for the samples with and without quality responses, respectively. Similarly, for the selected feature subsets $\tilde{\mathcal{F}}_B$, $\tilde{\mathcal{F}}_C$, and $\tilde{\mathcal{F}}_D$, we obtain matrices $\widetilde{\boldsymbol{X}}_B \in \mathbb{R}^{N \times \tilde{P}_B}$, $\widetilde{\boldsymbol{X}}_C \in \mathbb{R}^{N \times \tilde{P}_C}$, and $\widetilde{\boldsymbol{X}}_D \in \mathbb{R}^{N \times \tilde{P}_D}$. Each matrix is further split into two matrices: $\widetilde{\boldsymbol{X}}_{1B} \in \mathbb{R}^{N_1 \times \tilde{P}_B}$,$\widetilde{\boldsymbol{X}}_{1C} \in \mathbb{R}^{N_1 \times \tilde{P}_C}$, and $\widetilde{\boldsymbol{X}}_{1D} \in \mathbb{R}^{N_1 \times \tilde{P}_D}$ for the N_1 samples with

quality response; $\widetilde{\boldsymbol{X}}_{2B} \in \mathbb{R}^{N_2 \times \tilde{P}_B}$, $\widetilde{\boldsymbol{X}}_{2C} \in \mathbb{R}^{N_2 \times \tilde{P}_C}$, and $\widetilde{\boldsymbol{X}}_{2D} \in \mathbb{R}^{N_2 \times \tilde{P}_D}$ for those without quality response.

Table 3.4 Selected features by sparse group lasso

Tensor decomposition / SGL method / parameters	MPCA	UMPCA
(α^*, λ^*)	$\tilde{\mathcal{F}}_A$ = {F_1, PC_1, PC_3, AC/CA}	$\tilde{\mathcal{F}}_B$ = {F_1, PC_1, AC/CA}
$(\alpha = 0.95, \lambda^*)$	$\tilde{\mathcal{F}}_C$ = {F_1, PC_1, PC_3, AC/CA, υ, ω}	$\tilde{\mathcal{F}}_D$ = {F_1, PC_1, AC/CA, υ, ω}

3.6.2 Weighted Classifier Fusion Results

The individual classifiers in the proposed fusion approach are:

i. logistic regression,

ii. KSVM with polynomial kernel of degree $d = 2$,

iii. KSVM with Gaussian kernel of $\gamma = 0.25$ and non-separable penalty parameter of 100,

iv. neural network with 2 hidden layers,

v. *k*-nearest neighbors with k = 7.

The fusion model in Eq. (3.11) has $\pi_0 = 0.5$ as the decision boundary and $a = 1000$ as the convergence rate. As described in Section 3.5.2, cross-validation is used to determine the optimal weights in classifier fusion. Taking all N_1 samples into consideration, the optimal weights for each feature subset are shown Table 3.5. w_{lr}^*, w_{psvm}^*, w_{gsvm}^*, w_{nn}^* and w_{knn}^* are the optimal weights for logistic regression, polynomial KSVM, gaussian KSVM, neural network and *k*-nearest neighbors, respectively.

Table 3.5 Optimal weights for weighted classifier fusion

Scenario	Optimal weights / Dataset	w_{lr}^*	w_{psvm}^*	w_{gsvm}^*	w_{nn}^*	w_{knn}^*
A	$\widetilde{\boldsymbol{X}}_A$	0.8605	0.0000	0.0107	0.1161	0.0127
B	$\widetilde{\boldsymbol{X}}_B$	0.2461	0.0918	0.0761	0.2955	0.2905
C	$\widetilde{\boldsymbol{X}}_C$	0.2430	0.0009	0.0237	0.3543	0.3781
D	$\widetilde{\boldsymbol{X}}_D$	0.4243	0.1851	0.1834	0.0410	0.2072

The N_1 = 18 samples with quality response are divided into a training dataset and a testing dataset by stratified sampling. Specifically, the testing dataset contains four samples: two randomly selected from the low-quality group Ω_0 and two from the high-quality group Ω_1. Since Ω_0 contains 8 samples and Ω_1 contains 10 samples, the total number of unique datasets is $\binom{8}{2} \times \binom{10}{2}$ = 1260. The proposed method is applied to each unique partition to obtain a training CCR and a testing CCR. Considering all 1260 unique partitions, we obtain the $\overline{CCR}$, s_{CCR}, and $SNR = \overline{CCR}/s_{CCR}$. The results are shown in Table 3.6.

We have the following observations from Table 3.6. All scenarios have high $\overline{CCR}$ and small s_{CCR} for the training dataset. The highest training $\overline{CCR}$ is obtained in scenario D while the smallest training s_{CCR} is obtained in scenario C. Scenario C yields the highest training SNR. Scenarios B and D also give high SNR results for the training dataset. Scenario B yields the best result for the testing dataset with the highest $\overline{CCR}$ and smallest s_{CCR}. Therefore, we conclude that the feature subset in scenario B gives the best quality evaluation results.

Table 3.6 Quality evaluation results

Scenario	SGL parameters	Tensor decomposition method	**Training performance**		
			$\overline{CCR}$	s_{CCR}	SNR
A	(α^*, λ^*)	MPCA	0.9298	0.0651	14.2762
B	(α^*, λ^*)	UMPCA	0.8923	0.0428	20.8516
C	$(\alpha = 0.95, \lambda^*)$	MPCA	0.9385	0.0381	24.6338
D	$(\alpha = 0.95, \lambda^*)$	UMPCA	0.9477	0.0496	19.1132
Scenario	SGL parameters	Tensor decomposition method	**Testing performance**		
			$\overline{CCR}$	s_{CCR}	SNR
A	(α^*, λ^*)	MPCA	0.7017	0.2233	3.1426
B	(α^*, λ^*)	UMPCA	0.8313	0.1699	4.8934
C	$(\alpha = 0.95, \lambda^*)$	MPCA	0.7118	0.1863	3.8212
D	$(\alpha = 0.95, \lambda^*)$	UMPCA	0.6912	0.2084	3.3166

For the *in-situ NDE* of FSBR, we recommend extracting features by UMPCA, then selecting the optimal feature subset $\tilde{\mathcal{F}}_B$ = {F_1, PC_1, AC/CA}, and then determining the quality of the FSBR joints by weighted classifier fusion with optimal weights $\mathbf{w}^* = (0.2461, 0.0918, 0.0761, 0.2955, 0.2905)^T$. Based on these results, the quality prediction model in Eq. (3.11) can be rewritten as:

$$\hat{y}_i = \frac{1}{1 + \exp\left(-1000 \times \left[\left(\boldsymbol{w}^* \times h_c\left(\tilde{\boldsymbol{X}}_{B,i}\right)\right) - 0.5\right]\right)}, \tag{3.12}$$

where h_c represents the c^{th} individual classifier, c = 1, …, 5. $\tilde{\boldsymbol{X}}_{B,i}$ is the i^{th} row in matrix $\tilde{\boldsymbol{X}}_B$, representing the i^{th} sample in FSBR joints. $\hat{y}_i$ is the prediction label for the i^{th} sample.

$\hat{y}_i$ = 0 indicates that the i^{th} sample is classified into the low-quality group Ω_0, whereas $\hat{y}_i$ = 1 indicates that the i^{th} sample is classified into the high-quality group Ω_1.

The developed model is then applied to the FSBR samples that do not have quality response. Among the N_2 = 56 joints that were not tested, 48 are predicted as of high quality and 8 samples are predicted as of low quality.

3.7 Conclusion

The statistical monitoring of multiple sensor profiles in friction stir blind riveting represents a new challenging field of research in understanding the FSBR process and improving the overall performance of lightweight materials manufacturing. This chapter develops a process monitoring method to gain a better understanding of FSBR in joining carbon fiber-reinforced polymer and aluminum alloy sheets. The developed method establishes the statistical relationship between the FSBR process and the quality of the joints.

Our study investigates the effectiveness of the unsupervised tensor decomposition methods, particularly, generalizations of the basic PCA-based approach, in extracting features from multi-sensor, high-dimensional, heterogeneous profile data. These PC features are combined with process parameters, material stack-up sequence, and engineering-driven features such as the peak force to provide rich information about the FSBR process. Our study also explores the effectiveness of sparse group lasso regression in selecting optimal features from the extracted ones, considering the presence of both continuous and categorical variables. The selected features are fed into weighted classification fusion to estimate the quality of the FSBR joints. The fusion method integrates five individual classifiers and their weights are determined by optimizing the

classification performance in cross-validation. Quality evaluation is treated as a binary classification problem in this study. The available FSBR samples are clustered into a high-quality group and a low-quality group by hierarchical clustering, and their cluster labels are considered as the true quality group labels. The correct classification rates resulted from various feature extraction methods and feature selection results are assessed and compared. The results indicate that both MPCA and UMPCA are effective feature extraction methods. The average correct classification rate is more than 80%.

The proposed method fully utilizes online process information to evaluate the quality of the FSBR joints. It is expected that the proposed method will help reduce the time and efforts in quality inspection, especially destructive tests. By treating feed rate, spindle speed, and material stack-up sequence as categorical features, the proposed method can be easily extended to include other lightweight materials or more process parameters. The tensor decomposition methods for feature extraction is applicable to multi-sensor profile data, making the proposed method easily updated when new sensors are added to the system. The proposed method can also be applied to other discrete manufacturing processes with online sensing capabilities for the purpose of process monitoring and *in-situ* quality evaluation.

Further research efforts will be devoted to improving the quality evaluation model. For example, the proposed method will be extended to virtual metrology. In addition to classifying the samples into a quality group, virtual metrology will incorporate a regression model that enables the estimation of the maximum tensile load or even multiple key quality characteristics.

CHAPTER 4 OPTIMAL INTEGRATION OF SUPERVISED TENSOR DECOMPOSITION AND ENSEMBLE LEARNING FOR ENABLING *IN-SITU* NDE

When developing statistical methods for manufacturing problems, class information such as low/high quality may be provided for a limited number of training samples via experiments. This motivates us to consider supervised feature extraction methods that take class information into consideration. However, due to time and cost in conducting these experiments, usually only a small number of samples can be provided, resulting in a *small sample size* (SSS) scenario for supervised algorithms. In SSS, the training sample size is smaller than the data dimension, leading to non-exist or inaccurate evaluation results. Therefore, this chapter aims to improve *in-situ* NDE by developing a new sensor fusion and process monitoring method that is able to handle the SSS problem while considering available class information. To achieve this objective, we propose to integrate uncorrelated multilinear discriminant analysis and ensemble learning to monitor the manufacturing process and evaluate the quality of products.

Introduced in Lu *et al*. (2009), uncorrelated multilinear discriminant analysis (UMLDA) decomposes the tensor format data to low-dimensional format by maximizing the Fisher Discriminant Criterion (FDC), which is the ratio between the between-class scatters and the within-class scatters. The extracted features from UMLDA are mutually uncorrelated and this property ensures minimal redundancy in the extracted features, simplifying the classification task for quality evaluation. Because of the scatter ratio calculation in UMLDA, it suffers from the SSS problem. In SSS scenario, the within-class scatter matrix is ill-posed and the inverse of this matrix may not exist and as a

result, the optimal FDC ratio may not exist or be unbounded. To handle the SSS scenario, UMLDA is transformed into a regularized-UMLDA (R-UMLDA) method by adding a regularization term such that the within-scatter matrix is modified into a full rank matrix (Lu *et al*., 2009).

The performance of R-UMLDA heavily depends on the regularization parameter. Different feature matrices can be extracted when different regularization parameters are used. Hence, R-UMLDA cannot be considered as a stable feature extractor. This requires caution in handling this instability. As discussed in Chapter 2, ensemble learning is promising in handling unstable features and potentially making a weak classifier stronger. Therefore, we propose to integrate the R-UMLDA feature extractor with ensemble learning to stabilize the feature extraction and to increase the classification performance. Based on the review of various ensemble learning algorithms in Chapter 2, we adopt Bagging and Adaboost in this research.

It is pointed out in Chapter 2 that the performance of ensemble learning depends on many aspects. In this research, we focus on diversity manipulation. Generally speaking, the higher the diversity among base learners is, the better performance an ensemble learning algorithm can provide. As reviewed in Chapter 2, both Bagging and Adaboost algorithms manipulate their base learner diversity in the sample space. Bagging generates diversity via the repeatable but selective bootstrapped sampling method; Adaboost develops the base learner diversity by adding different modified weights on different training samples, based on their performance in the previous iteration. We propose a new diversity manipulation method for ensemble learning that not only keeps the original diversity in the sample space but also introduces additional diversity from the

feature space. The proposed method is expected to provide optimal integration of R-UMLDA feature extractor and ensemble learning algorithms, taking the advantages of both methods.

In the proposed method, multiple regularization parameters are selected from a wide range of the possible values and the corresponding feature matrices are obtained. Next, the extracted feature matrices are clustered into a number of feature groups and the group centers are chosen as the most diversified feature matrices. Clustering is adopted to maximize the diversity among feature matrices in different groups and minimize the diversity within feature matrices in same groups. We suggest the number of feature groups to be 20 to strike a balance between the range of the regularization parameter and the computational complexity. Then, we concatenate these 20 selected matrices into a single matrix as the input for ensemble learning. In Lu *et al*. (2009), 20 regularization parameters were selected uniformly from $[10^{-7}, 10^{-2}]$ in log scale for 20 R-UMLDA extractors. This arbitrary value selection neglects the possible dependence between the regularization parameter and the input data. We develop a heuristic adaptive method to find an optimal interval for the regularization parameters and the selected regularization parameters are expected to provide the optimal *in-situ* quality evaluation result.

The proposed method is compared to competitors in both the numerical studies and FSBR case study. Competitors include R-UMLDA with aggregation (R-UMLDA-A) and nearest neighbor classifier (Lu *et al*., 2009) and R-UMLDA with the standard Bagging or Adaboost, without the diversity optimization. The performances of these methods are evaluated in the correct classification rate (CCR), same as in Chapter 3.

4.1 Method Overview

The proposed method is illustrated in Figure 4.1. In Step 1, the input tensor data $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_M}$ is decomposed via R-UMLDA based on regularization parameter $\gamma, \gamma \geq 0$, and the known class label vector $\boldsymbol{y}$. The feature extraction result is denoted as feature matrix $\boldsymbol{F} \in \mathbb{R}^{I_M \times k}$, where I_M is the number of observations and k represents number of extracted features. Since R-UMLDA is sensitive to γ and a single feature matrix is not robust enough for the subsequent classification task, tensor $\mathcal{A}$ is decomposed with multiple R-UMLDA feature extractors, each with a different regularization parameter in $\boldsymbol{\gamma} = [\gamma_1, \cdots, \gamma_n]$; the corresponding extracted feature matrices are $\boldsymbol{F}_1, \cdots, \boldsymbol{F}_n$.

In Step 2, to enhance the ensemble learning performance, we develop a novel diversity-based optimal feature selection method via clustering to manipulate the base learner diversity from the feature space. In this step, the n extracted feature matrices are clustered into 20 groups and the group centers are adopted as the most diversified feature matrices, denoted as $\boldsymbol{F}_1^*, \cdots, \boldsymbol{F}_{20}^*$. We define the diversity among feature matrices as the distance among these matrices in two-dimensional space.

In Step 3, the selected feature matrices are concatenated into a single matrix $\boldsymbol{F}^* = [\boldsymbol{F}_1^*, \cdots, \boldsymbol{F}_{20}^*]$. Then $\boldsymbol{F}^*$ is fed into Bagging and Adaboost to obtain the generalized classification error $\hat{e}$. Decision tree is adopted as the base learner. The classification error $\hat{e}$ is expressed as

$$\hat{e} = \frac{|\hat{\boldsymbol{y}} - \boldsymbol{y}|^0}{N}, \tag{4.1}$$

where the operation $|\cdot|^0$is the L_0 norm to count the different elements between $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}$; $\boldsymbol{y} \in \mathbb{R}^{N \times 1}$ is the true class label vector of the testing observations in tensor $\mathcal{A}$; $\hat{\boldsymbol{y}} \in \mathbb{R}^{N \times 1}$

is the predicted class label vector from the proposed method; N is the number of testing observations.

In Step 4, we develop a heuristic adaptive partition approach to find a proper interval $[a, b]$ for the regularization parameter γ. The initial interval is divided into several narrower subintervals and for each iteration i, n parameters $\boldsymbol{\gamma}^i = \{\gamma_1^i, \cdots, \gamma_n^i\}$ are selected from the subinterval $[a^i, b^i]$ uniformly in log scale and the 20 group centers $\boldsymbol{F}_1^*, \cdots, \boldsymbol{F}_{20}^*$ are selected according to Step 2, next the corresponded regularization parameters $\boldsymbol{\gamma}^{*i} = \{\gamma_1^{*i}, \cdots, \gamma_{20}^{*i}\}$ to the selected group centers can be determined. However, these 20 selected regularization parameters may not be uniformly located in log scale any more. The generalization error rate $\hat{e}^i$ is calculated according to Step 3. Once the $\hat{e}^i$ is smaller than the generalization error rate in the previous iteration $\hat{e}^{i-1}$, the search continues and another 20 parameters $\boldsymbol{\gamma}^{*(i+1)} = \{\gamma_1^{*(i+1)}, \cdots, \gamma_{20}^{*(i+1)}\}$ from the next subinterval $[a^{i+1}, b^{i+1}]$ are selected. This adaptive approach improves the method performance with moving-window subintervals. Once $\hat{e}^i$ is larger than $\hat{e}^{i-1}$, we assume that the optimal error rate $\hat{e}^* = \hat{e}^{i-1}$ is obtained and the corresponding subinterval $[a^{i-1}, b^{i-1}]$ is the optimal interval $[a^*, b^*]$. The number of subintervals should be chosen such that the computational complexity can be handled.

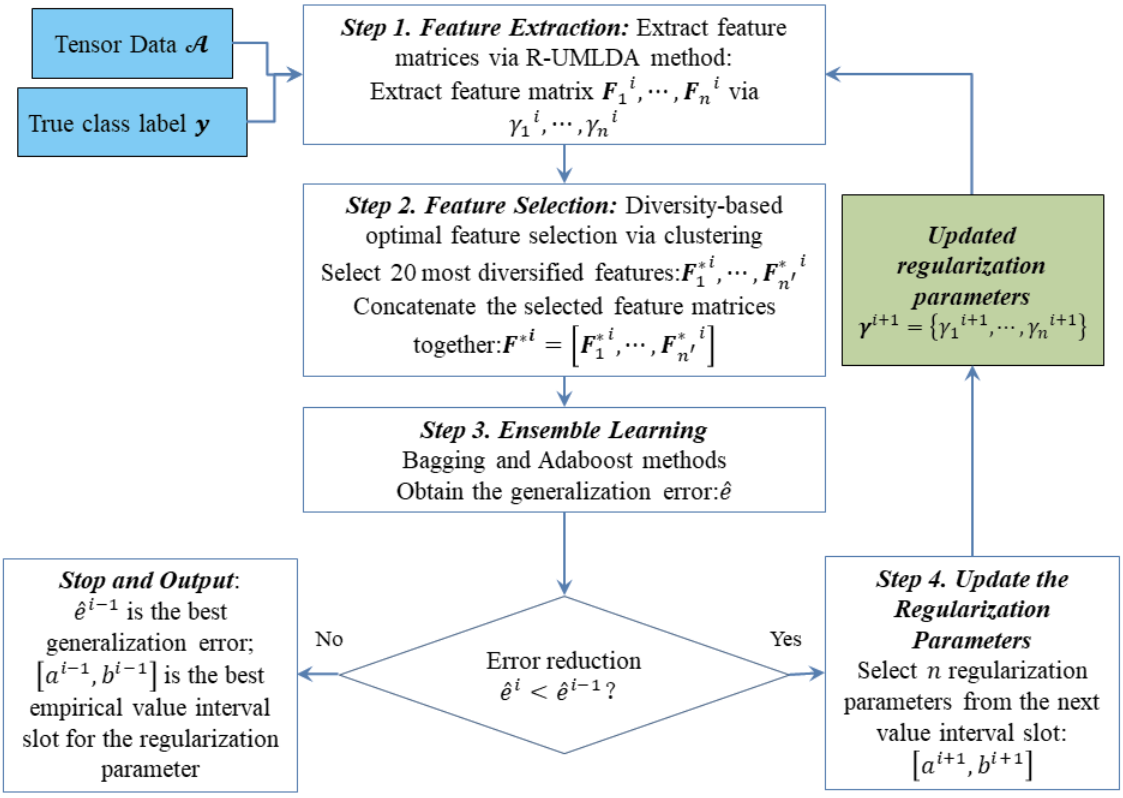


Figure 4.1 Flowchart of the proposed method

To sum up, our proposed method consists of four steps: Step 1, feature extraction via R-UMLDA; Step 2, diversity-based optimal feature selection via clustering; Step 3, ensemble learning; and Step 4, updating regularization parameters. Among the four steps, Step 2 is the core step to integrate feature extraction and ensemble learning. Step 2 utilizes the advantages of the uncorrelated property of the R-UMLDA algorithm and adds additional diversity from the feature space. Based on the error-ambiguity decomposition for ensemble learning algorithms (Chen, 2008), we prove that increasing diversity among base learners can reduce the generalization error for Bagging and Adaboost M1; the same observation can be made for Adaboost M2 empirically.

4.2 Regularized UMLDA for Feature Extraction

As a multilinear extension of the regular LDA method, UMLDA extracts features from the tensor space directly via a series of multilinear operations of TVP (tensor to vector projection) to maximize the scatter-ratio-based criterion. Like the regular LDA method, UMLDA also suffers from the SSS problem, under which the inverse of the within-class scatter matrix may not exist. To handle the SSS scenario, regularization is added to stable the within-class scatter matrix during the TVP process. Thus, UMLDA is modified to R-UMLDA method and the feature extraction performance is dependent on the selection of the regularization parameter γ.

4.2.1 The R-UMLDA Approach

Using the same multilinear algebra notations presented in Chapter 3, input data $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is a 3-mode tensor. UMLDA applies the TVP to map the tensor object into uncorrelated vectors (the extracted features). The TVP projects a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ to a P-dimensional vector via P EMPs (elementary multilinear projections). Each EMP transfers the tensor $\mathcal{A}$ into a scalar z by $z = \langle \mathcal{A}, \mathcal{U} \rangle$, where $\mathcal{U}$ is the projection tensor with each mode as the projection vectors. Without loss of generality, the third mode in tensor $\mathcal{A}$ can be set as the sample size and $\mathcal{A}_m$ is denoted as the m^{th} sample of the two-dimensional tensor object, $\mathcal{A}_m \in \mathbb{R}^{I_1 \times I_2}$. Hence, the p^{th} EMP projection can be represented as:

$$z_{m_p} = \langle \mathcal{A}_m, \mathcal{U} \rangle = \mathcal{A}_m \times {}_1\boldsymbol{u}_p^{1\,T} \times {}_2\boldsymbol{u}_p^{2\,T}, \tag{4.2}$$

where z_{m_p} is the projected scalar; $\boldsymbol{u}_p^{1\,T}$ and $\boldsymbol{u}_p^{2\,T}$ are projection vectors along modes I_1 and I_2, respectively, $p = 1, \cdots, P$. To calculate the scatter ratio criterion for UMLDA, the

within-class scatter and the between-class scatter under the p^{th} projection can be denoted as S_{w_p} and S_{B_p}:

$$S_{w_p} = \sum_{m=1}^{M} \left(z_{m_p} - \bar{z}_{c_{m_p}} \right)^2 \text{ and } S_{B_p} = \sum_{c=1}^{C} M_c \left(\bar{z}_{c_p} - \bar{z}_p \right)^2, \tag{4.3}$$

where C is the number of classes; M_c is the sample size in the c^{th} class, $c = 1, \cdots, C$; c_{m_p} is the class label of the m^{th} sample under the p^{th} projection; M is the total sample size; $\bar{z}_p = (1/M) \sum_{m=1}^{M} z_{m_p}$ is the mean of all the projected samples and can be set as $\bar{z}_p = 0$ to simplify the calculation. $\bar{z}_{c_p} = (1/M_c) \sum_{m=1, c_m = c} z_{m_p}$ is the mean of the projected scalars in class c. Thus, the scatter ratio under the p^{th} projection is $F_p = S_{B_p} / S_{w_p}$. Set $\boldsymbol{g}_p$ as the p^{th} coordinate vector, with its m^{th} element $g_p(m) = z_{m_p}$. The coordinate vector in UMDLA method guarantees the zero-correlation between the two projected vectors. Hence, the objective of the UMLDA is to find P EMPs $\left\{ \boldsymbol{u}_p^{1\,T}, \boldsymbol{u}_p^{2\,T} \right\}_{p=1,\cdots,P}$ that maximize the scatter ratio of F_p and ensure the projected vectors with zero-correlations. The objective function to find the P EMPs is:

$$\begin{gathered} \left\{ \boldsymbol{u}_p^{1\,T}, \boldsymbol{u}_p^{2\,T} \right\}_{p=1,\cdots,P} = \operatorname{argmax} F_p \\ \text{Subject to } \left\| \boldsymbol{u}_p^1 \right\| = 1, \left\| \boldsymbol{u}_p^2 \right\| = 1, \\ \frac{\mathbf{g}_p{}^T \mathbf{g}_q}{\|\mathbf{g}_p\| \|\mathbf{g}_q\|} = \delta_{pq}, p, q = 1, \cdots, P. \end{gathered} \tag{4.4}$$

where δ_{pq} is the kronecker delta, defined as $\delta_{pq} = 1$, when $p = q$ and $\delta_{pq} = 0$ otherwise; operation $\|\cdot\|$ is the Euclidean norm. Solution of the Eq. (4.4) follows the successive determination approach and the detail to compute these EMPs can be found in Lu *et al*. (2009).

If the training sample size is smaller than the data dimension, UMLDA suffers from the small sample size (SSS) problem, under which the optimization process always pushes the within-class scatter S_{w_p} to 0. The consequences are that the inverse of S_{w_p} may not exist and the optimal scatter ratio F_p becomes unbounded. Regularization is introduced in Lu *et al*. (2009) with regularization parameter $\gamma > 0$ to address this issue and the UMLDA is then revised to R-UMLDA. The solution procedure of R-UMLDA follows the same procedure as UMLDA. During the p^{th} projection, given $\mathcal{A}_m \in \mathbb{R}^{I_1 \times I_2}$, R-UMLDA assumes one projection vector $\left\{\boldsymbol{u}_p^{1\,T} \text{or } \boldsymbol{u}_p^{2\,T}\right\}$ is known and the 2-mode tensor $\mathcal{A}_m \in \mathbb{R}^{I_1 \times I_2}$ is projected to a vector $\tilde{\boldsymbol{z}}_p^l = \mathcal{A}_m \times \boldsymbol{u}_p^l$, $l = 1$ or 2. After introducing the regularization parameter γ, the within-class scatter matrix $\tilde{\boldsymbol{S}}_{w_p}^l$ is calculated as:

$$\tilde{\boldsymbol{S}}_{w_p}^l = \sum_{m=1}^{M} \left(\tilde{\boldsymbol{z}}_{m_p}^l - \bar{\tilde{\boldsymbol{z}}}_{c_{m_p}}^l\right)\left(\tilde{\boldsymbol{z}}_{m_p}^l - \bar{\tilde{\boldsymbol{z}}}_{c_{m_p}}^l\right)^T + \gamma \cdot \lambda_{max}\left(\breve{\boldsymbol{S}}_w^l\right) \cdot \boldsymbol{I}_{I_l}, \tag{4.5}$$

where $\boldsymbol{I}_{I_l}$ is the identity matrix with size $I_l \times I_l$; $\bar{\tilde{\boldsymbol{z}}}_{c_{m_p}}^l = (1/M_c)\sum_{m=1,c_m=c} \tilde{\boldsymbol{z}}_{m_p}^l$ is the mean of the projected results under the p^{th} projection for class c. $\lambda_{max}\left(\breve{\boldsymbol{S}}_w^l\right)$ is the largest eigen-value of $\breve{\boldsymbol{S}}_w^l$, which is the within-class scatter matrix for the l-mode vectors of the training tensor samples and the $\breve{\boldsymbol{S}}_w^l$ is defined as:

$$\breve{\boldsymbol{S}}_w^l = \sum_{m=1}^{M} \left(\boldsymbol{A}_m^l - \bar{\boldsymbol{A}}_{c_m}^l\right)\left(\boldsymbol{A}_m^l - \bar{\boldsymbol{A}}_{c_m}^l\right)^T, \tag{4.6}$$

where $\bar{\boldsymbol{A}}_{c_m}^l$ is the l-mode unfolded matrix of the class mean tensor $\bar{\mathcal{A}}_c = (1/M_c)\sum_{m,c_m=c} \mathcal{A}_m$. After introducing the regularization parameter, the within-class scatter matrix $\tilde{\boldsymbol{S}}_{w_p}^l$ is fully ranked and the scatter ratio F_p is bounded during the maximization and therefore the R-UMLDA method can cover the small training sample

size scenario. The solution procedure can be easily adopted to tensors with mode higher than 2.

It should be noticed that $\tilde{\boldsymbol{S}}_{w_p}^{l}$ is an approximated estimation of the true within-class scatter matrix and this estimation is scaled by $\cdot \lambda_{max}\left(\tilde{\boldsymbol{S}}_{w}^{l}\right) \cdot \boldsymbol{I}_{I_l}$, which is controlled by the regularization parameter γ. The feature extraction sensitivity to γ can be transferred to the subsequent classification process to generate larger bias and variance in the classification results. Researchers have tried to mitigate this issue to provide a more reliable feature extraction result for the subsequent classification process. R-UMLDA aggregation (R-UMLDA-A) with the NNC (nearest neighbor classifier) algorithm is such an attempt.

4.2.2 R-UMLDA with Aggregation

R-UMLDA Aggregation (R-UMLDA-A) with nearest neighbor classifier (NNC) has been developed to mitigate the drawbacks of using a single R-UMLDA feature extractor. R-UMLDA-A can achieve an aggregated classification performance by summing up the normalized individual results from the single R-UMLDA feature extractors with the NNC classifier. To classify a test tensor $\mathcal{A}$, R-UMLDA-A algorithm projects this tensor to different feature groups $\boldsymbol{F}_1, \cdots, \boldsymbol{F}_n$ first via n individual R-UMLDA extractors. Then, the nearest neighbor distance for the i^{th} R-UMLDA extractor from the test tensor $\mathcal{A}$ to all the class center $\bar{\mathcal{A}}_c, c = 1, \cdots, C$ is calculated. This distance is represented as $\text{dis}[\mathcal{A}, \bar{\mathcal{A}}_c] = \min_c \|\boldsymbol{F}_i - \overline{\boldsymbol{F}}_c\|$, where the operation of $\|\cdot\|$ is the Euclidean distance measurement, $\boldsymbol{F}_i$ is the i^{th} extracted feature matrix and $\overline{\boldsymbol{F}}_c$ is the centered extracted feature matrix for class c. Next, this Euclidean distance is normalized and the test tensor $\mathcal{A}$ is assigned to class label $c^* = \text{argmin}_c\, d[\mathcal{A}, c]$, where d is the summation of the normalized distances for all

the n R-UMLDA feature extractors. Details of the R-UMLDA-A algorithm can be found in Lu *et al*. (2009).

R-UMLDA-A usually outperforms individual R-UMLDA. The superiority of R-UMLDA-A can be explained in view of ensemble learning. R-UMLDA-A algorithm can be regarded as a special ensemble learning algorithm belonging to the "mixture of experts" family. R-UMLDA-A takes NNC as the base learners and trains them with different input data matrices $\boldsymbol{F}_1, \cdots, \boldsymbol{F}_n$; the results are then combined via the sum rule. Since the extracted feature matrices are different, R-UMLDA-A can provide good diversity among the base learners and hence greatly strengthen the classification performance. Drawbacks of R-UMLDA-A can also be explained in view of ensemble learning. First, the aggregated performance of R-UMLDA-A is almost always better when more R-UMLDA extractors are aggregated and as a result, R-UMLDA-A may easily be overfitting if too many individual R-UMLDA extractors are combined. Second, performance of R-UMLDA-A is also strongly dependent on γ. Therefore, choosing the regularization parameter γ should consider the different feature matrices, the risk of overfitting, and the potential computational complexity. This issue is addressed in the proposed method.

4.3 Optimal Integration of R-UMLDA and Ensemble Learning

We propose an effective integration of R-UMDLA feature extraction and ensemble learning. This integration aims to utilize the advantages from both methods. On one hand, R-UMLDA can enhance the ensemble learning performance by first enlarging the feature space and then optimizing the feature selection via diversity manipulation. On the other hand, ensemble learning can stabilize the R-UMLDA feature extraction and help determine the optimal values for the regularization parameter in R-UMLDA. In this

section, we first describe the proposed ensemble learning performance reinforcement via diversity manipulation on the input features. We then present a new heuristic adaptive method to optimize the regularization parameter selection for R-UMLDA. Two widely used ensemble learning algorithms are discussed in this section: Bagging and Adaboost. For Adaboost, Adaboost M1 and Adaboost M2 are adopted to handle binary classification and multi-class classification, respectively. Details of these algorithms can be found in Breiman (1996) and Schapire (2003).

4.3.1 Diversity of Base Learners

It is widely believed that the diversity and accuracy of the base learners are critical to the effectiveness of ensemble learning, demonstrated by theoretical analysis and empirical studies (Zhou, 2012). Generally speaking, a higher diversity encourages a better ensemble learning performance. Our proposed method introduces additional diversity to the original ensemble learning algorithms to obtain a better performance.

There are existing studies on how base learners in ensemble learning would affect regression performance. The impact can be explained via the error-ambiguity decomposition (Krogh and Vedelsby, 1995) and the bias-variance-covariance decomposition (Geman *et al.*, 1992). However, there is no literature on evaluating the diversity of the base learners or how this diversity affects the classification performance.

The difficulty to generate such an evaluation is from many aspects that covers almost all parts of the ensemble learning classification. In this research, we analyze the base learner diversity by error-ambiguity decomposition and we explain the quantitative relationship between the diversity and the generalization error. Our analysis is based on the binary classification task, assuming class labels as the outputs from the individual and

the ensembled classifiers, using $0-1$ loss function and the weighted average combining rule.

Adopting a similar derivation process as in Chen (2008), $\boldsymbol{F}=\begin{pmatrix}\boldsymbol{f^1}\\ \vdots \\ \boldsymbol{f^M}\end{pmatrix}$ is the input data matrix, where $\boldsymbol{f}^i$ is the i^{th} sample, $i=1,\cdots,M$ and $y_i=\pm 1$ is the true class label of the i^{th} sample. $G_j, j=1,\cdots,T$ is defined as the j^{th} individual base classifier and $G_j(\boldsymbol{f}^i)=\pm 1$ is the predicted class label for the i^{th} sample from the j^{th} base classifier. G_{en} is the ensembled classifier. Under the weighted average combining rule, $G_{en}(\boldsymbol{f}^i)=\sum_{j=1}^{T}\alpha_j G_j(\boldsymbol{f}^i)$ and the label for the i^{th} sample is assigned as $\tilde{G}_{en}(\boldsymbol{f}^i)=\text{sign}[G_{en}(\boldsymbol{f}^i)]$, where $\text{sign}(\cdot)$ is the sign function; $\alpha_j>0$ and $\sum_{j=1}^{T}\alpha_j=1$ are the weights for base classifiers. $G_{en}(\boldsymbol{f}^i)\in[-1,1]$ and $\tilde{G}_{en}(\boldsymbol{f}^i)=\pm 1$.

Using the margin theory, we define $\rho_{ens}^i=y_i G_{en}(\boldsymbol{f}^i)$ as the margin for the ensembled classifier on sample i and $\rho_{ens}^i\in[-1,1]$; $\rho_j^i=y_i G_j(\boldsymbol{f}^i)$ is the margin for the j^{th} base classifier on sample i and $\rho_j^i=\pm 1$. Next, we define the error function $Err(\cdot)$ as

$$Err(x)=\begin{cases}1, if\ x=-1\\ 0, if\ x=1\end{cases}, \tag{4.7}$$

which can be written as $Err(x)=-\frac{1}{2}(x-1)$, where x can be treated as the margin ρ_{ens}^i or ρ_j^i. Therefore, $Err(\cdot)$ is a monotonic decreasing function of the margins. For the i^{th} sample, the generalization error from the ensembled classifier G_{en} is denoted as $Err(\rho_{ens}^i)$; the generalization error from the j^{th} individual base classifier of G_j is denoted as $Err(\rho_j^i)$; the Error-Ambiguity decomposition can be expressed as:

$$Err(\rho_{ens}^i)=\sum_{j=1}^{T}\alpha_j Err(\rho_j^i)\ -\mathrm{A}\,, \tag{4.8}$$

where A is defined as the ambiguity among base classifiers. From Eq. (4.8), the ambiguity for the i^{th} sample can be expressed as:

$$\mathrm{A} = \frac{y_i}{2}\sum_{j=1}^{T}\left[\frac{1}{T}G_{ens}(\boldsymbol{f}^i) - \alpha_j G_j(\boldsymbol{f}^i)\right]. \tag{4.9}$$

The term $\sum_{j=1}^{T}\alpha_j G_j(\boldsymbol{f}^i)$ in Eq. (4.9) can be written as $\sum_{j=1}^{T}\alpha_j G_j(\boldsymbol{f}^i) = \left|\sum_{j=1}^{T}\alpha_j G_j(\boldsymbol{f}^i)\right| \operatorname{sign}\left[\sum_{j=1}^{T}\alpha_j G_j(\boldsymbol{f}^i)\right] = \mathrm{s}\tilde{G}_{en}(\boldsymbol{f}^i)$, where s ∈ [0,1] is the absolute value of $\sum_{j=1}^{T}\alpha_j G_j(\boldsymbol{f}^i)$ and $\tilde{G}_{en}(\boldsymbol{f}^i) = \operatorname{sign}\left[\sum_{j=1}^{T}\alpha_j G_j(\boldsymbol{f}^i)\right]$ assigns the class label to sample i. Thus, the term A can be expressed as:

$$\mathrm{A} = \frac{y_i}{2}\left[\tilde{G}_{en}(\boldsymbol{f}^i) - sG_{en}(\boldsymbol{f}^i)\right], \tag{4.10}$$

and the Error-Ambiguity decomposition is therefore converted to:

$$Err(\rho_{ens}^i) = \sum_{j=1}^{T}\alpha_j Err(\rho_j^i) - \frac{y_i}{2}\left[\tilde{G}_{en}(\boldsymbol{f}^i) - sG_{en}(\boldsymbol{f}^i)\right], \tag{4.11}$$

where $\tilde{G}_{en}(\boldsymbol{f}^i) = \begin{cases} 1, G_{en}(\boldsymbol{f}^i) \geq 0 \\ -1, G_{en}(\boldsymbol{f}^i) \leq 0 \end{cases}$.

Since a positive ensemble learning margin which implies a lower generalized error is encouraged, we can always assume $\rho_{ens}^i > 0$. Based on $\rho_{ens}^i = y_i G_{en}(\boldsymbol{f}^i)$ and $y_i = \pm 1$, we can always set $y_i = \tilde{G}_{en}(\boldsymbol{f}^i)$.

Case 1: If $G_{en}(\boldsymbol{f}^i) \geq 0$, $y_i = \tilde{G}_{en}(\boldsymbol{f}^i) = 1$. Eq. (4.11) is converted into $(\rho_{ens}^i) = \sum_{j=1}^{T}\alpha_j Err(\rho_j^i) + \frac{1}{2}sG_{en}(\boldsymbol{f}^i) - \frac{1}{2}$, which is a monotonic increasing function of s.

Case 2: If $G_{en}(\boldsymbol{f}^i) \leq 0$, $y_i = \tilde{G}_{en}(\boldsymbol{f}^i) = -1$. Eq. (4.11) is converted into $(\rho_{ens}^i) = \sum_{j=1}^{T}\alpha_j Err(\rho_j^i) - \frac{1}{2}sG_{en}(\boldsymbol{f}^i) - \frac{1}{2}$, which is also a monotonic increasing function of s.

From Case 1 and Case 2, we observe that a smaller s can reduce the ensemble learning error $Err(\rho_{ens}^i)$. Since $\mathrm{s} = \left|\sum_{j=1}^{T}\alpha_j G_j(\boldsymbol{f}^i)\right|$ measures the difference between the

positive and negative votes from the individual classifiers, a smaller s encourages larger diversity among base learners, based on the diversity definition in Section 2.3.1. In this way, increasing the diversity among individual classifiers can reduce the generalization error of the ensembled classification.

The above analysis fits Adaboost M1 and Bagging very well. The major difference between Bagging and Adaboost is in the combining rules. The majority voting method $G_{ens}(\boldsymbol{f}^i) = \text{argmax}_{y\in\{1,-1\}} \sum_{j=1}^{T} \mathbb{I}\left[G_j(\boldsymbol{f}^i) = y_i\right]$ is the combining rule used in Bagging algorithm, where $\mathbb{I}(\cdot)$ is the indicator function and $\sum_{j=1}^{T} \mathbb{I}\left[G_j(\boldsymbol{f}^i) = y_i\right]$ counts number of individual classifiers that predict the class label as $y_i = \pm 1$. For the binary class task, the expression $\text{argmax}_{y\in\{1,-1\}} \sum_{j=1}^{T} \mathbb{I}\left[G_j(\boldsymbol{f}^i) = y_i\right]$ is equivalent to $\text{sign}\left[\sum_{j=1}^{T} \alpha_j G_j(\boldsymbol{f}^i)\right]$, which is the weighted average combining rule in Adaboost algorithms, given equal base classifier weights of $\alpha_j = \frac{1}{T}, j = 1, \cdots, T$. Since $G_j(\boldsymbol{f}^i) = \pm 1$, i.e., once k base classifiers, $k > \frac{T}{2}$, provide the class label 1 or -1 as the outputs, the function $\text{argmax}_{y\in\{1,-1\}} \sum_{j=1}^{T} \mathbb{I}\left[G_j(\boldsymbol{f}^i) = y_i\right]$ assigns 1 or -1 as the final output of the bagging method. This is the same result as the assigned ensemble class label from Adaboost M1 via function $\text{sign}\left[\sum_{j=1}^{T} \alpha_j G_j(\boldsymbol{f}^i)\right]$. Therefore, increasing the diversity among base classifiers can increase the classification performance of ensemble learning for both Bagging and Adaboost M1.

This theoretical derivation cannot cover Adaboost M2, which is the multiclass version of Adaboost. Empirical study has shown the diversity increase among base classifiers can also reduce the generalization error for Adaboost M2.

4.3.2 Improving Diversity in Feature Space

We adopt decision tree as the based learner in our proposed method. Therefore, the construction of the base classifiers is equivalent to building individual trees. Our proposed method increases the diversity among base learners according to decision tree. The standard Bagging and Adaboost algorithms with decision tree as base learner generate diversity from the sample space. Different from these original diversities, we introduce additional diversity from the feature space. This extra diversity utilizes the different feature matrices obtained from different R-UMLDA extractors on the same input tensor $\mathcal{A}$ via different regularization parameters.

A decision tree is a tree-structured classifier that assigns the class label to an observation in a divide-and-conquer way (Breiman, 2017; Zhou, 2012). In each step, the decision tree splits the dataset into different parts according to different results associated to certain feature test. This split continues until a classification decision is made. The feature test in each split step can be conducted by applying the information gain criterion which measures the entropy reduction of the dataset $\boldsymbol{F} = [\boldsymbol{F}_1, \cdots, \boldsymbol{F}_{20}]$, $\boldsymbol{F_i} \in R^{M \times k}, i = 1, \cdots, 20$. If the splitting decision is made from the feature $\boldsymbol{a}_i, i = 1, \cdots, 20k$, the information gain criterion can be expressed as:

$$IG(\boldsymbol{F}, \boldsymbol{a}_i) = H(\boldsymbol{F}) - H(\boldsymbol{F}|\boldsymbol{a}_i), \tag{4.12}$$

where $H(\boldsymbol{F})$ is the entropy prior to the splitting and $H(\boldsymbol{F}|\boldsymbol{a}_i)$ is the entropy after the splitting at feature $\boldsymbol{a}_i$. Eq. (4.12) can be written as:

$$IG(\boldsymbol{F}, \boldsymbol{a}_i) = -\sum_{j=1}^{C} p_j \log_2 p_j - \sum_{i=1}^{20k} p(\boldsymbol{a}_i) \sum_{j=1}^{C} [-p(j|\boldsymbol{a}_i) \log_2 p(j|\boldsymbol{a}_i)], \tag{4.13}$$

where $j = 1, \cdots, C$ are all the possible classes in $\boldsymbol{F}$; p_j is the proportion of observations in class j and $\sum_{j=1}^{C} p_j \log_2 p_j$ is the entropy $H(\boldsymbol{F})$ prior to the splitting on feature $\boldsymbol{a}_i$; $p(\boldsymbol{a}_i)$

can be regarded as the weight of the feature $\boldsymbol{a}_i$ and equal weight is assumed in our proposed method; $p(j|\boldsymbol{a}_i)$ is the proportion of observations grouped in class j after splitting on feature $\boldsymbol{a}_i$ and $\sum_{j=1}^{C}[-p(j|\boldsymbol{a}_i)\log_2 p(j|\boldsymbol{a}_i)]$ is the entropy $H(F|\boldsymbol{a}_i)$. The feature $\boldsymbol{a}_i^*$ where the split decision is made is the feature that maximizes the information gain:

$$\boldsymbol{a}_i^* = \text{argmax}_{\boldsymbol{a}_i, i=1,\cdots,20k}\ IG(\boldsymbol{F}, \boldsymbol{a}_i)\,. \tag{4.14}$$

Empirically, a feature with high mutual information is preferred in Eq. (4.14). Therefore, the selected feature $\boldsymbol{a}_i^*$ should contain as much similar information as possible to others. However, from Eq. (4.11), a diversified base learner committee is preferred for ensemble learning generalization performance. Therefore, to increase the diversity among individual trees, we need to reduce the mutual information shared in different features, otherwise the split points from different trees might be the same or very similar and the individual trees are not as much different as proposed. Since R-UMLDA extracts uncorrelated features in the feature matrix, increasing the diversity among features is equivalent to increasing the difference among feature matrices. From this point, we utilize the uncorrelated property of R-UMLDA features and introduce a clustering-based diversity maximization method to increase the mutual diversity among individual trees.

4.3.3 Diversity-based Optimal Feature Selection via Clustering

After n feature matrices $\boldsymbol{F}_1, \cdots, \boldsymbol{F}_n$ are extracted via the regularization parameter $\boldsymbol{\gamma} = [\gamma_1, \cdots, \gamma_n]$ from the same input tensor $\mathcal{A}$, assuming n is much larger than 20, we propose a clustering-based distance measurement to group the n extracted matrices into 20 groups and select the center of these groups as the most diversified feature matrices $\boldsymbol{F}_1^*, \cdots, \boldsymbol{F}_{20}^*$. The proposed method clusters the feature matrices by maximizing the feature matrix

differences in different groups and minimizing the feature matrix differences in the same groups. Define $\boldsymbol{S} = \{S_1, \cdots, S_{20}\}$ as the 20 groups, the objective function to find the groups:

$$\boldsymbol{S}^* = \text{argmin}_{\boldsymbol{S}} \sum_{i=1}^{20} \sum_{\substack{\boldsymbol{F}_j \in \boldsymbol{S}_i \\ j=1,\cdots,n}} \left\| \boldsymbol{F}_j - \boldsymbol{\mu}_i \right\|, \tag{4.15}$$

where $\boldsymbol{\mu}_i$ is the mean of the feature matrices in $\boldsymbol{S}_i$ and $\boldsymbol{\mu}_i$ is defined as the center of set i; the extracted feature matrices that represented by $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_{20}$ are the most diversified feature matrices $\boldsymbol{F}_1^*, \cdots, \boldsymbol{F}_{20}^*$; operation $\|\cdot\|$ is the distance measurement, we adopt the distance correlation method as our distance measurement. Since all the n feature matrices are extracted from the same tensor $\mathcal{A}$, the distance correlation method can evaluate their differences directly: a smaller correlation implies a larger mutual difference.

It should be noted that the optimal process to group n observations into 20 groups is an NP-complete problem (Friedman *et al*., 2001). Therefore, the total number n cannot be set very large. In both the numerical and the case studies, we recommend $n = 250$ and the average number that can be grouped in each set is around 12~13. k-means clustering method, which selects the arithmetic center of $\boldsymbol{S}_i$ as the cluster center $\boldsymbol{\mu}_i$, is recommended in our method. Empirically, the samples that are grouped in each cluster may be very few and the arithmetic center may not represent any true extracted feature matrix $\boldsymbol{F}_i, i = 1, \cdots, n$. Under this situation, the feature matrix that is closest to the arithmetic center will be nominated as the group center. The k-median clustering method, which sets the rank center of $\boldsymbol{S}_i$ as the cluster center $\boldsymbol{\mu}_i$, is also considered in our method.

4.3.4 Optimal Regularization Parameters for Integration

The regularization parameter γ is the only parameter in R-UMLDA and it controls the feature extraction performance in terms of the generalized classification error. It is

necessary to optimize this parameter or provide recommendations for empirical application. Since γ can be any positive values, $\gamma > 0$, it is computationally impossible to enumerate all the feasible solutions. In Lu *et al*. (2009), an empirical interval of $[10^{-7}, 10^{-2}]$ is suggested and the γ's are selected from this interval uniformly in log scale. In this research, we develop a heuristic adaptive method to find an optimal interval $[a, b]$ for γ in terms of classification performance without significantly increasing the computational complexity.

As shown in Eq. (4.5), the estimated within-class scatter $\tilde{\boldsymbol{S}}_{w_p}^l$ is the sum of true within-class scatter $\boldsymbol{S}_{w_p}^l$ and the regularization term $\gamma \cdot \lambda_{max}(\breve{\boldsymbol{S}}_w^l) \cdot \boldsymbol{I}_{I_l}$. According to singular value decomposition, $\boldsymbol{S}_{w_p}^l = \sum_{i=1}^{I_l} \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^T$, where λ_i is the i^{th} eigenvalue of $\boldsymbol{S}_{w_p}^l$ and $\boldsymbol{v}_i$ is the corresponded eigenvector. Since $\boldsymbol{S}_{w_p}^l$ is usually ill-posed under the small sample size scenario, many eigenvalues are close to or equal to 0. In the ratio-based Fisher discriminant criterion $F_p = \boldsymbol{S}_{B_p}^l {\boldsymbol{S}_{w_p}^l}^{-1}$, the inverse of the true within-class scatter matrix ${\boldsymbol{S}_{w_p}^l}^{-1} = \sum_{i=1}^{I_l} \frac{\boldsymbol{v}_i \boldsymbol{v}_i^T}{\lambda_i}$ can be unbounded or very unstable. Eq. (4.5) converts the term ${\boldsymbol{S}_{w_p}^l}^{-1}$ into:

$$ {\tilde{\boldsymbol{S}}_{w_p}^l}^{-1} = \sum_{i=1}^{I_l} \frac{\boldsymbol{v}_i \boldsymbol{v}_i^T}{\lambda_i + \gamma \cdot \lambda_{max}(\breve{\boldsymbol{S}}_w^l)}, \tag{4.16} $$

where $\lambda_{max}(\breve{\boldsymbol{S}}_w^l)$ is defined as the largest eigenvalue of the within-class scatter matrix $\breve{\boldsymbol{S}}_w^l$ along the l mode for all the training samples and this term is scaled by the regularization parameter γ. Therefore, the estimated within-class scatter matrix $\tilde{\boldsymbol{S}}_{w_p}^l$ is dependent on both the input data through the eigenvalue $\lambda_i, i = 1, \cdots, I_l$, and γ. The γ value should be carefully chosen such that regularization is effective. If γ is too large comparing to the

λ_i's, $\gamma \cdot \lambda_{max}(\tilde{\boldsymbol{S}}_w^l)$ will dominate the denominator in Eq. (4.16) and the estimation of $\boldsymbol{S}_{w_p}^l$ will be highly biased. On the other hand, a small γ value (close to zero) will make the regularization ineffective.

This phenomenon motivates us to consider the generalized classification error $\hat{e}$ as a convex function of $\boldsymbol{\gamma} = \{\gamma_1, \cdots, \gamma_n\}$ in [0,1]. When $\boldsymbol{\gamma}$ are chosen from an interval $[a, b]$ that is close to 0, $\hat{e}$ will be large. By moving the interval $[a, b]$ from close to 0 to close to 1, the error will decrease first and then increase. Therefore, the minimum error $\hat{e}^*$ can be found by choosing $\boldsymbol{\gamma}$ from an optimal interval $[a^*, b^*]$.

We develop an adaptive interval partition method based on the above analysis. This is a heuristic method to find the optimal interval for $\boldsymbol{\gamma}$. Initially, $\boldsymbol{\gamma}$ can be uniformly selected in log scale from an interval close to 0, for example, $[10^{-10}, 10^{-5}]$. We then move the interval from $[10^{-10}, 10^{-5}]$ to $[10^{-5}, 1]$, one decimal magnitude per iteration. The generalization classification errors $\hat{e}_1, \cdots, \hat{e}_6$ can be calculated according to our proposed method via Step 3 in Figure 4.1. The classification errors are compared in sequence and the adaptive interval partition procedure can be terminated once an increase in $\hat{e}_i$ is observed. The optimal error is $\hat{e}^* = \hat{e}_{i-1}$ and the corresponding interval $\left[a^{i-1}, b^{i-1}\right]$ is the optimal interval $[a^*, b^*]$ for $\boldsymbol{\gamma}$. Since the number of subintervals is not very large, our adaptive method increases the computational complex in polynomial time.

It should be noticed that although $\boldsymbol{\gamma} = \{\gamma_1, \cdots, \gamma_n\}$ is initially uniformly selected from the subintervals in log scale, the final selected $\boldsymbol{\gamma}^* = \{\gamma_1^*, \cdots, \gamma_{20}^*\}$ may not be uniformly located. This is because, following the proposed method, $\boldsymbol{\gamma}^*$ corresponds to the clustered group centers $\boldsymbol{F}_1^*, \cdots, \boldsymbol{F}_{20}^*$, which may not be uniformly located in log scale any more.

4.4 Numerical Study

In this section, we conduct numerical studies to evaluate the proposed optimal integration method. R-UMLDA-A is set as the benchmark method. The naive integration of R-UMLDA and ensemble learning is denoted as R-UMLDA-EL. We compare the proposed method with R-UMLDA-EL to demonstrate the impact of introducing extra diversity from the feature space. The proposed method is evaluated in two datasets: multi-profile signal data and image data.

4.4.1 Multi-profile Signal Data

Based on the numerical study in Grasso *et al*. (2014) and Guo *et al*. (2016), we simulate the multi-profile signal data with benchmark signals proposed in Donoho and Johnstone (1994). The benchmark signals have been widely used to test wavelet-based algorithms and applied in research field of statistical learning and data sciences (Grasso *et al*., 2014). Following the same setup as in Guo *et al*. (2016), three benchmark signals are used: blocks, heavysine, and bump. R-UMLDA methods allow capturing the major signal features without any further modeling or smoothing. The three benchmark signals are denoted as $\boldsymbol{x_1}$,$\boldsymbol{x_2}$ and $\boldsymbol{x}_3$ and shown in Figure 4.2.

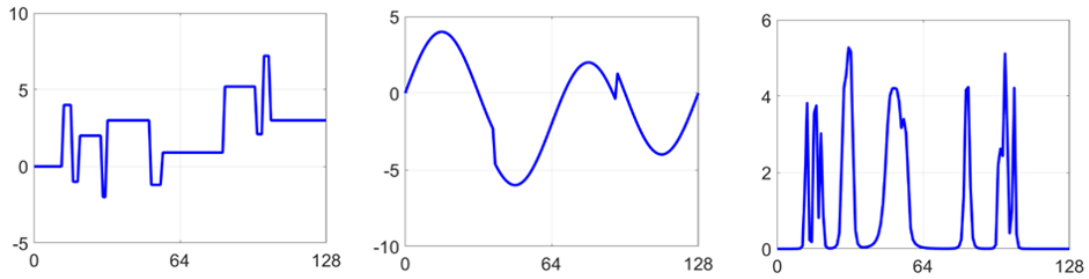


Figure 4.2 Benchmark signals: 'blocks', 'heavysine' and 'bumps'

Following the same setup as in Guo *et al*. (2016), we let $\chi \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ denote a three-mode tensor to represent the simulated multi-profile signals from four sensors, where $I_1 = 4$ is the number of sensors, $I_2 = 128$ is the signal length and $I_3 = M$ represents the simulated sample size. The four signals are generated following Eq. (4.17):

$$\begin{aligned} \chi_{1,.,m} &= \beta_{1,m}\boldsymbol{x}_1 + \beta_{2,m}\boldsymbol{x}_2 + \varepsilon_{1,m} \\ \chi_{2,.,m} &= \beta_{3,m}\boldsymbol{x}_1{}^2 + \beta_{4,m}\boldsymbol{x}_3 + \varepsilon_{2,m} \\ \chi_{3,.,m} &= \beta_{5,m}\boldsymbol{x}_2{}^2 + \beta_{6,m}\boldsymbol{x}_3{}^2 + \varepsilon_{3,m} \\ \chi_{4,.,m} &= \beta_{7,m}\boldsymbol{x}_1\boldsymbol{x}_2 + \varepsilon_{4,m} \end{aligned} \tag{4.17}$$

where $\varepsilon_{n,m} \sim N(0, 0.5^2), n = 1, \cdots, 4$ and $m = 1, \cdots, M$ is the random noise and is set the same for all the four multi-profile signals; $\boldsymbol{B} = [\beta_{1.m}, \cdots, \beta_{7.m}]^T \sim MVN(\boldsymbol{\mu}_B, \boldsymbol{\Sigma}_B)$ is the model parameter vector; $\boldsymbol{\mu}_B = [0.2,1,1.5,0.5,1,0.7,0.8]^T$ and $\boldsymbol{\Sigma}_B = \text{diag}(\sigma_{B1}^2, \cdots, \sigma_{B7}^2) = \text{diag}(0.08,0.015,0.05,0.01,0.09,0.03,0.06)$. In these 3-mode tensors, we generate the multi-profile signals with as complete signal correlations as possible to simulate the complex conditions that may be faced in real applications. It can be noticed that the signal correlations that exist in our simulation include the linear correlations (e.g., $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ in $\chi_{1,.,m}$; $\boldsymbol{x}_3$ in $\chi_{2,.,m}$), the second order curvilinear correlations (e.g., $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ in $\chi_{3,.,m}$; $\boldsymbol{x}_1$ in $\chi_{2,.,m}$) and without correlations (e.g., $\boldsymbol{x}_3$ with $\chi_{4,.,m}$).

We consider one in-control scenario and five out-of-control scenarios in this simulation. The five out-of-control scenarios cover five possible anomaly conditions that may happen in joining processes. The anomalous magnitudes are defined as $\delta_1, \delta_2, \delta_3, \delta_4, \delta_5$, respectively.

- Mean shift of the 'blocks' signal: $\boldsymbol{x}_1' = \boldsymbol{x}_1 + 0.1\delta_1 \mathbf{1}_{128\times1}$, where $\mathbf{1}_{128\times1}$ is a column vector of ones.
- Superimposition of a sinusoid term with the 'block' signal: $\boldsymbol{x}_1'' = \boldsymbol{x}_1 + 0.1\delta_2 \boldsymbol{y}_s$, where $\boldsymbol{y}_s$ is a sine function over the domain $[0, I_2]$ with the period I_2 and peak-to-peak amplitude equal to 1.
- Standard deviation increment for the error term: $\varepsilon_{1,m}' = \delta_3 \varepsilon_{1,m}$.
- Mean shift of μ_{B1} , the first element in $\boldsymbol{\mu}_B$: $\mu_{B1}' = \mu_{B1} + \delta_4 \sigma_{B1}$, where ${\beta_{1,m}}' \sim N(\mu_{B1}', {\sigma_{B1}}^2)$.
- Standard deviation increment for the model parameter $\sigma_{B1}' = \delta_5 \sigma_{B1}$, where ${\beta_{1,m}}'' \sim N\left(\mu_{B1}, {\sigma_{B1}'}^2\right)$.

Among the five out-of-control scenarios, signals 1, 2 and 4 are affected in (a) and (b); signal 1 is affected in (c), (d), and (e); signal 3 is not affected in this simulation at all. 11 anomalous magnitude groups are assigned to these five scenarios to represent different anomalous severities from trivial noise to significant changes. The 11 magnitude groups for the anomalous conditions are listed in Table 4.1.

Table 4.1 Magnitude of the signal changes

Change	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*
δ_1	0.001	0.010	0.020	0.030	0.040	0.050	0.060	0.070	0.0800	0.090	0.100
δ_2	0.025	0.035	0.045	0.055	0.065	0.075	0.085	0.095	0.105	0.115	0.125
δ_3	1	1.2	1.4	1.6	1.8	2.0	2.2	2.4	2.6	2.8	3.0
δ_4	1	1.4	1.8	2.2	2.6	3.0	3.4	3.8	4.2	4.6	5.0
δ_5	1.5	1.7	1.9	2.1	2.3	2.5	2.7	2.9	3.1	3.3	3.5

We generate 200 samples in each scenario. An example of the simulated signals with the change magnitude of $[\delta_1, \delta_2, \delta_3, \delta_4, \delta_5] = [0.1, 0.125, 3.0, 5.0, 3.5]$ is displayed in Figure 4.3.

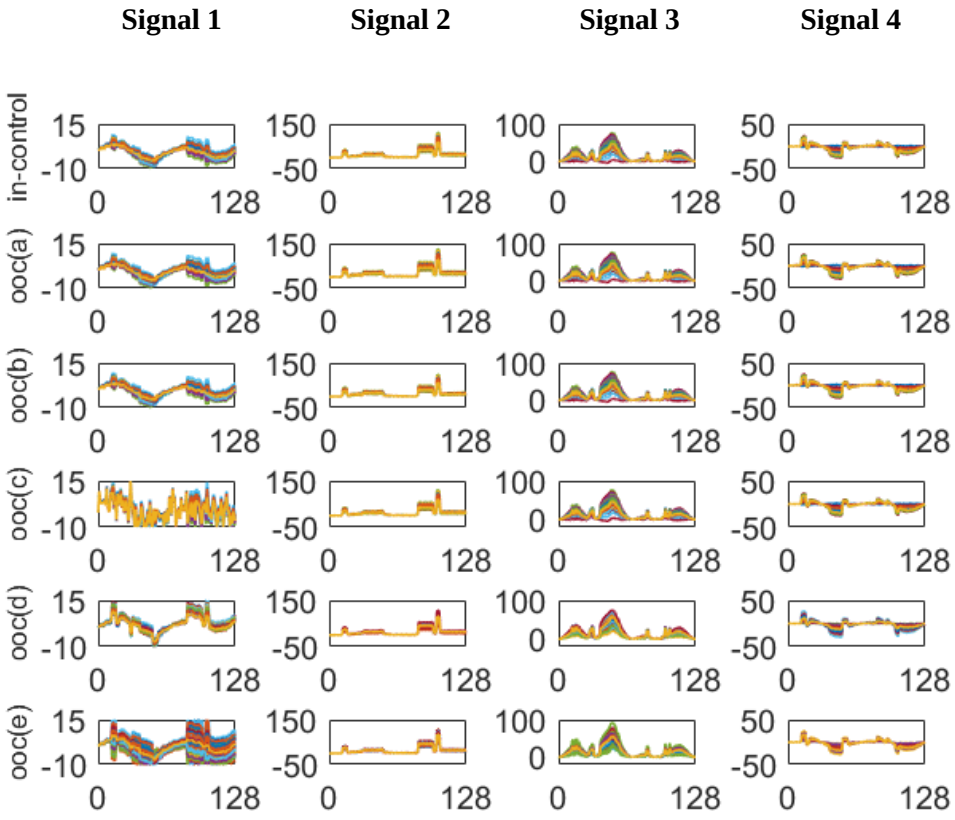


Figure 4.3 Simulated dataset, 1200 samples in 6 classes with the change magnitude of $[\delta_1 = 0.1, \delta_2 = 0.125, \delta_3 = 3.0, \delta_4 = 5.0, \delta_5 = 3.5]$

4.4.2 Image Data

Based on the same image generation method in Yan *et al*. (2015), we consider the symmetric $60 \times 60 \times 3$ RGB image with red foreground circle on the black background as the baseline in-control scenario. Four out-of-control scenarios are generated to represent foreground changes in location, size, shape, and color, respectively. Each individual image is a three-mode tensor $\phi \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, where $I_1 = I_2 = 60$ store the image pixel information along the horizontal and vertical directions; I_3 is the mode for the R, G, B color channel. Adding the sample size, we obtain a four-mode tensor $\phi' \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4}$ with the I_4 representing the number of images. The simulated images are generated via three sets of parameters: location of the foreground circle center, circle radius, and the RGB color. All five scenarios use the same background. $\boldsymbol{\delta} = [\delta_1, \delta_2, \delta_3, \delta_4]^T$ represents the magnitudes of changes in the four out-of-control scenarios. We use the same setup as in Yan *et al*. (2015) to generate the images but modify the magnitudes to better demonstrate our method:

(a) In-control image: the center parameter is a two dimensional random vector that follows multivariate normal distribution $\left[L_{0x}, L_{0y}\right]^T \sim MVN(\boldsymbol{\mu}_{0L}, \boldsymbol{\Sigma}_{0L})$, where $\boldsymbol{\mu}_{0L} = \left[\mu_{0x}, \mu_{0y}\right]^T = [30, 30]^T$ and $\boldsymbol{\Sigma}_{0L} = \text{diag}\left[\sigma_{0x}^2, \sigma_{0y}^2\right]^T = \text{diag}[1,1]^T$; parameter of circle radius is a constant vector $\boldsymbol{R}_0 = \left[r_{0x}, r_{0y}\right]^T = [10,10]^T$; color parameter (RGB numbers) is also generated by following the multivariate normal distribution as $[c_{01}, c_{02}, c_{03}]^T \sim MVN(\boldsymbol{\mu}_{0c}, \boldsymbol{\Sigma}_{0c})$, where $\boldsymbol{\mu}_{0c} = [\mu_{0R}, \mu_{0G}, \mu_{0B}]^T = [100, 20,15]^T$ and $\boldsymbol{\Sigma}_{0c} = \text{diag}[\sigma_{0R}^2, \sigma_{0G}^2, \sigma_{0B}^2] = \text{diag}[10, 2,1.5]$.

(b) Location change: the mean of the center is shifted to $\boldsymbol{\mu}_{1L} = \left[\mu_{1x}, \mu_{1y}\right]^T = \left[\mu_{0x}, \mu_{0y}\right]^T + \delta_1\left[\sigma_{0x}, \sigma_{0y}\right]^T$.

(c) Radius change: the circle radius is changed to $\boldsymbol{R}_2 = (1+\delta_2)\boldsymbol{R}_0 = \left[r_{2x}, r_{2y}\right]^T = \left[(1+\delta_2)r_{0x}, (1+\delta_2)r_{0y}\right]^T$.

(d) Shape change: the circle is changed to an ellipsis with $\boldsymbol{R}_3 = \left[r_{3x}, r_{3y}\right]^T = \left[(1+\delta_3)r_{0x}, \frac{r_{0y}}{1+\delta_3}\right]^T$. The new foreground shape in this scenario is dependent on the aspect ratio of the generated ellipses, which is denoted as $s = \frac{r_{3x}}{r_{3y}} = (1+\delta_3)^2$.

(e) Color change: the foreground color is shifted to $\boldsymbol{\mu}_{4c} = [\mu_{4R}, \mu_{4G}, \mu_{4B}]^T = [\mu_{0R}, \mu_{0G}, \mu_{0B}]^T + \delta_4[\sigma_{0R}, \sigma_{0G}, \sigma_{0B}]^T$.

We generate 200 samples in each class. Total 10 change magnitude groups are assigned to the out-of-control scenarios and the value of the changes are displayed in Table 4.2. An example of the change group is shown in Figure 4.4. It should be noted that one can hardly recognize the changes in Figure 4.4 via visual inspection.

Table 4.2 Magnitudes of the image changes

Changes	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
δ_1	0.15	0.30	0.45	0.60	0.75	0.90	1.05	1.20	1.35	1.50
δ_2	0.005	0.010	0.015	0.020	0.025	0.030	0.035	0.040	0.045	0.050
δ_3	0.005	0.010	0.015	0.020	0.025	0.030	0.035	0.040	0.045	0.050
δ_4	0.005	0.010	0.015	0.020	0.025	0.030	0.035	0.040	0.045	0.050

Original Image | ***Location Change*** $\boldsymbol{\delta_1 = 1.5}$ | ***Size Change*** $\boldsymbol{\delta_2 = 0.05}$ | ***Shape Change*** $\boldsymbol{\delta_3 = 0.05}$ | ***Color Change*** $\boldsymbol{\delta_4 = 0.05}$

Figure 4.4 In-control and out-of-control images

4.4.3 Dataset Partition

We consider two dataset partition strategies for both the multi-profile signal data and the image data to simulate two situations commonly occurred in real applications. The first strategy is to select 5 samples from each class to construct the training dataset and all other samples are used for testing. This strategy is designed to mimic the small sample size scenario that our method is specifically designed for in this research. This strategy represents many processes where only a limited amount of data is available due to possible time and cost constraints. The second strategy is to distribute the simulated data from each class evenly into the training dataset and the testing dataset, i.e., 100 samples per class for the two datasets. The second dataset partition strategy represents processes where data collection can be easily performed. The two dataset partition strategies are listed in Table 4.3.

Table 4.3 Dataset partition under the numerical study

		Sample size	
		Multi-profile signal data	Image data
Partition 1	Training dataset	30	600
	Testing dataset	1170	600
Partition 2	Training dataset	25	500
	Testing dataset	975	500

4.4.4 Methods in Comparison

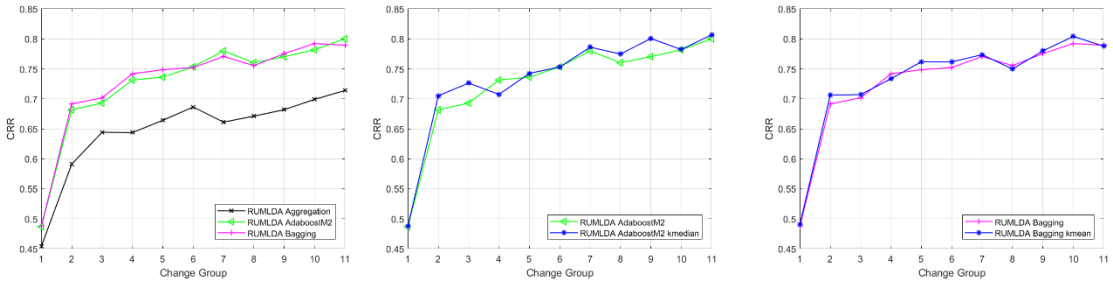
We compare the proposed method with the R-UMLDA-A method as well as two naïve R-UMLDA-EL methods, using Bagging and Adaboost, respectively. We denote these two R-UMLDA-EL methods as R-UMLDA-Bagging and R-UMLDA-Adaboost M1/M2. The regularization parameter $\boldsymbol{\gamma} = \{\gamma_1 \cdots, \gamma_{20}\}$ are uniformly selected from $[10^{-7}, 10^{-2}]$ in log scale for R-UMLDA-A and R-UMLDA-EL while $\boldsymbol{\gamma}$ in the proposed method are selected according to the following steps: (1) determine 6 subintervals in $[10^{-10}, 1]$ and each subinterval covers six decimal magnitudes, e.g., the first subinterval is $[10^{-10}, 10^{-5}]$ and the last subinterval is $[10^{-5}, 1]$; (2) select $\boldsymbol{\gamma} = \{\gamma_1, \cdots, \gamma_{250}\}$ uniformly from each subinterval in log scale; (3) select optimal features according to Section 4.3.3 and obtain the corresponding group centers $\boldsymbol{F}^{*\boldsymbol{i}} = \left[\boldsymbol{F}_1^{*\,i}, \cdots, \boldsymbol{F}_{20}^{*\,i}\right]$ and regularization parameters $\boldsymbol{\gamma}^{*i} = \left\{\gamma_1^{*\,i}, \cdots, \gamma_{20}^{*\,i}\right\}$ for iteration $i, i = 1, \cdots, 6$, respectively; (4) find the optimal generalization performance $\hat{e}^*$ and the corresponding $\boldsymbol{\gamma}^*$ and optimal subinterval. Since $\boldsymbol{\gamma}^*$ in the proposed method may vary in replicates, we will report the optimal subinterval $[a^*, b^*]$with the correct classification rate ccr^* for each simulated data set. The optimal correct classification rate is calculated as:

$$ccr^* = 1 - \hat{e}^*\,, \tag{4.18}$$

where $\hat{e}^*$ is defined in Section 4.3.4 and calculated by Eq. (4.1). We consider both the k-means and k-median clustering methods to maximize the diversity among the extracted feature matrices.

4.4.5 Simulation Results

Figure 4.5 shows the performance of our proposed method versus the competitors in terms of the average CCR. Each marker represents the average result from 10 simulation replicates for each magnitude group.



(a) **(b)** **(c)**

Figure 4.5 Average testing CCR for multi-profile signal data, 100 training samples per class: (a) R-UMLD-A vs. R-UMLDA-EL, (b) R-UMLDA Adaboost M2 vs. proposed method, (c) R-UMLDA-Bagging vs. proposed method

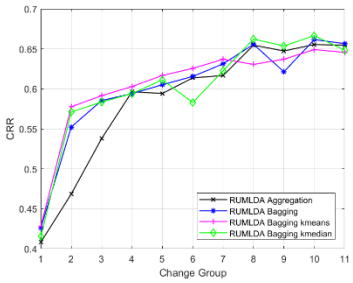


Figure 4.6 Average testing CCR for multi-profile signal data, 5 training samples per class, proposed method vs. competitors

Figure 4.5 displays the average testing CCR for the multi-profile signal data given 100 training samples per class. In Figure 4.5 (a), the black curve with cross markers corresponds to the average testing CCR of R-UMLDA-A while the red curve and the green curve represent the results from R-UMLDA-Bagging and R-UMLDA-Adaboost M2, respectively. It can be noticed that given 100 training samples per class, R-UMLDA with ensemble learning algorithms perform better than R-UMLDA-A. In Figure 4.5 (b), we compare the R-UMLDA-Adaboost M2 with our proposed method by using k-median diversity maximization method. Except for the 4^{th} magnitude group, our method outperforms R-UMLDA-Adaboost M2 in all the other groups; Figure 4.5 (c) shows the comparison result between R-UMLDA-Bagging and our method by using k-means diversity maximization method. Our method outperforms R-UMLDA-Bagging in all the magnitude groups except the 4^{th} group. Figure 4.6 displays the average testing CCR for all the competing methods given 5 training samples per class. Results from Adaboost M2 are not plotted in Figure 4.6 since Adaboost M2 in this situation reports an average CCR equal to random guess.

In Figure 4.6, we observe that R-UMLDA-Bagging with k-means diversity maximization method performs the best in all the magnitude groups except for groups 8, 9, and 10 while R-UMLDA-Bagging with k-median method wins in these three groups. Therefore, from Figure 4.5 and 4.6, we can conclude that our proposed method works better than the benchmark method and generally better than the naïve integration of R-UMLDA and ensemble learning. The extra diversity introduced in the proposed method plays a significant role in improving the overall classification performance. Table 4.4 lists

the average and standard deviation of the testing CCR for multi-profile signal data; the best average testing CCR are in bold.

Table 4.4 (a) Testing CCR (Mean±Std%): multi-profile signal data with 100 training samples per class

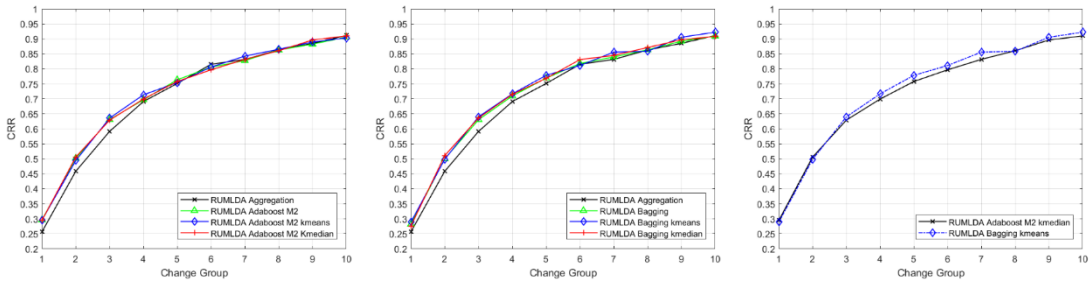
Change	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*
R-UMLDA - A	0.4543 ± 2.6	0.5912 ± 5.0	0.6443 ± 6.6	0.6438 ± 6.6	0.6642 ± 5.0	0.6865 ± 5.1	0.6612 ± 3.1	0.6712 ± 2.3	0.6820 ± 3.5	0.6992 ± 4.3	0.7140 ± 5.0
R-UMLDA Bagging	0.4875 ± 2.3	0.6917 ± 5.1	0.7017 ± 6.3	**0.7418 ± 5.4**	0.7487 ± 2.8	0.7522 ± 5.7	0.7707 ± 4.8	0.7555 ± 5.8	0.7755 ± 4.9	0.7923 ± 4.1	0.7895 ± 5.7
R-UMLDA Bagging *k*-means	**0.4905 ± 1.3**	0.7065 ± 2.4	0.7072 ± 7.4	0.7335 ± 5.0	**0.7620 ± 5.3**	**0.7617 ± 4.8**	0.7737 ± 4.2	0.7498 ± 5.2	0.7800 ± 6.0	**0.8042 ± 5.4**	0.7880 ± 4.3
R-UMLDA Bagging *k*-median	0.4768 ± 2.3	**0.7073 ± 4.6**	0.6942 ± 5.4	0.7262 ± 5.3	0.7577 ± 2.6	0.7580 ± 3.0	0.7642 ± 4.6	0.7412 ± 5.6	0.7727 ± 6.1	0.7747 ± 4.7	0.7992 ± 4.6
R-UMLDA AdaboostM2	0.4867 ± 2.0	0.6818 ± 4.4	0.6930 ± 6.3	0.7315 ± 4.5	0.7362 ± 5.7	0.7535 ± 4.6	0.7800 ± 2.9	0.7603 ± 3.7	0.7705 ± 3.0	0.7818 ± 4.2	0.8002 ± 3.0
R-UMLDA AdaboostM2 *k*-means	0.4770 ± 1.6	0.7070 ± 4.4	0.6928 ± 5.4	0.7253 ± 4.6	0.7630 ± 2.0	0.7592 ± 3.0	0.7722 ± 4.1	0.7448 ± 4.8	0.7788 ± 4.9	0.7713 ± 4.9	0.7995 ± 4.3
R-UMLDA AdaboostM2 *k*-median	0.4875 ± 2.4	0.7052 ± 2.0	**0.7263 ± 1.8**	0.7075 ± 4.1	0.7425 ± 4.9	0.7532 ± 4.1	**0.7862 ± 2.8**	**0.7748 ± 3.1**	**0.8007 ± 2.5**	0.7823 ± 3.6	**0.8068 ± 2.7**

Table 4.4 (b) Testing CCR (Mean±Std%): multi-profile signal data with 5 training samples per class

Change	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*
R-UMLDA - A	0.4083 ± 6.3	0.4685 ± 5.4	0.5379 ± 3.2	0.5964 ± 5.1	0.5943 ± 3.0	0.6138 ± 3.6	0.6167 ± 3.0	0.6547 ± 4.4	0.6474 ± 2.8	0.6555 ± 3.5	0.6547 ± 4.6
R-UMLDA Bagging	0.4262 ± 6.3	0.5523 ± 7.1	0.5855 ± 8.0	0.5941 ± 4.4	0.6052 ± 4.9	0.6156 ± 4.2	0.6313 ± 4.2	0.6557 ± 4.5	0.6214 ± 4.0	0.6616 ± 3.8	**0.6566 ± 5.1**
R-UMLDA Bagging *k*-means	**0.4282 ± 5.7**	**0.5776 ± 5.4**	**0.5915 ± 6.4**	**0.6029 ± 4.4**	**0.6168 ± 4.2**	**0.6256 ± 6.0**	**0.6370 ± 5.4**	0.6306 ± 4.4	0.6371 ± 3.9	0.6492 ± 4.1	0.6456 ± 2.9
R-UMLDA Bagging *k*-median	0.4155 ± 6.5	0.5710 ± 6.0	0.5836 ± 5.6	0.5942 ± 4.1	0.6112 ± 2.4	0.5829 ± 4.1	0.6227 ± 4.4	**0.6622 ± 3.7**	**0.6538 ± 3.4**	**0.6665 ± 3.6**	0.6532 ± 3.5

The same comparison is performed for the image dataset. Figure 4.7 shows the classification results given 100 training samples per class, where each marker represents the average result of 10 simulation runs for each magnitude group. The comparison between Adaboost M2 based methods and R-UMLDA-A is shown in Figure 4.7 (a); the comparison between Bagging based methods and R-UMLDA-A is shown in Figure 4.7 (b). Since it can be noticed that the Adaboost M2 with k-median and the Bagging with k-means outperform other methods in Figure 4.7 (a) and 4.7 (b), respectively, we further

compare these two methods in Figure 4.7 (c). The dotted line with diamond markers shows that the R-UMLDA Bagging with k-means method is better than R-UMLDA Adaboost M2 with k-median and therefore it is the best method for image data with 100 training sample size per class.



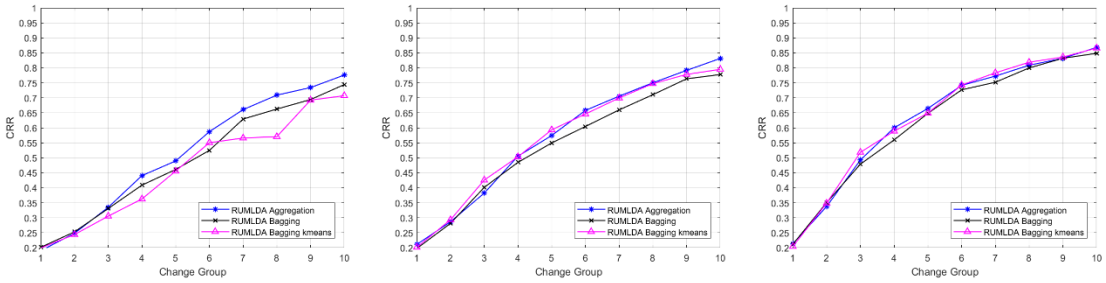
(a) **(b)** **(c)**

Figure 4.7 Average testing CCR for image data, 100 training samples per class: (a) R-UMLDA-A vs. R-UMLDA Adaboost M2 based methods, (b) R-UMLDA-A vs. R-UMLDA-Bagging based methods, (c) Comparison between two best proposed methods

The classification performance for the small sample size scenario is shown in Figure 4.8. Figure 4.8 (a), (b), and (c) show the average testing CCR given 5, 10 and 20 training samples per class, respectively. In these three figures, the blue, black, and red curves represent the performance of R-UMLDA-A, R-UMLDA Bagging, and R-UMLDA Bagging with k-means, respectively. Since Adaboost M2 cannot provide a better CCR than random guess, its results are not displayed here. Different from the curves in Figure 4.6 for the simulated multi-profile signal data, given 5 training samples per class, the benchmark method R-UMLDA-A has the best performance, R-UMLDA Bagging ranks the second and our proposed method cannot rival to the competitors. With the increment of the training sample size per class, the performance of our proposed method catches up and starts to overtake the competitors after 20 samples per class.

The different performance between the two simulated data sets can be explained from the extra dimension in the image data. The image data is a four way tensor with size of $\phi' \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4}$ while the multi-profile signal data is a three way tensor with size of $\chi \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ and hence four projection vectors $\{\boldsymbol{u}_1^T, \boldsymbol{u}_2^T, \boldsymbol{u}_3^T, \boldsymbol{u}_4^T\}$ should be estimated under R-UMLDA extractor for the image data, one more projection is need comparing to the multi-profile signal data. This extra estimation may reduce the recognition accuracy given very limited training samples. The average and standard deviation of the testing CCR for the image data are shown in Table 4.5.

The optimal subinterval of $\boldsymbol{\gamma}^*$ for the two numerical studies are the same: $[10^{-7}, 10^{-2}]$, which is the same interval used in Lu *et al*. (2009). However, in the proposed method, $\boldsymbol{\gamma}^* = [\gamma_1^*, \cdots, \gamma_{20}^*]$ is not uniformly located in log scale any more.



(a) **(b)** **(c)**

Figure 4.8 Average testing CCR for image data: (a) 5 training samples per class; (b) 10 training samples per class; (c) 20 training samples per class

Table 4.5 (a) Testing CCR (Mean±Std%): image data with 100 training samples per class

Change	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
R-UMLDA - A	0.2566 ± 1.7	0.4586 ± 1.9	0.5916 ± 1.8	0.6912 ± 2.0	0.7514 ± 1.7	0.8154 ± 2.3	0.8316 ± 1.7	0.8632 ± 1.4	0.8852 ± 1.5	0.9114 ± 1.5
R-UMLDA Bagging	0.2824 ± 2.7	0.5002 ± 1.9	0.6306 ± 2.0	0.7098 ± 1.9	0.7694 ± 2.5	0.8180 ± 2.4	0.8398 ± 1.9	0.8638 ± 1.4	0.8922 ± 2.1	0.9098 ± 1.3
R-UMLDA Bagging *k*-means	0.2892 ± 2.0	0.4986 ± 2.8	**0.6402 ± 2.9**	**0.7168 ± 1.8**	**0.7786 ± 1.5**	0.8114 ± 1.5	**0.8560 ± 1.6**	0.8586 ± 2.6	**0.9056 ± 1.1**	**0.9224 ± 1.2**
R-UMLDA Bagging *k*-median	0.2762 ± 1.7	**0.5110 ± 2.4**	0.6364 ± 1.9	0.7152 ± 1.7	0.7686 ± 2.5	**0.8312 ± 1.4**	0.8448 ± 1.4	**0.8714 ± 1.5**	0.8972 ± 1.2	0.9084 ± 1.2
R-UMLDA AdaboostM2	**0.2966 ± 3.0**	0.5020 ± 2.4	0.6308 ± 1.9	0.6972 ± 1.9	0.7636 ± 1.4	0.8058 ± 1.3	0.8278 ± 1.8	0.8630 ± 1.9	0.8826 ± 2.0	0.9060 ± 1.4
R-UMLDA AdaboostM2 *k*-means	0.2956 ± 2.5	0.4942 ± 3.1	0.6358 ± 1.8	0.7136 ± 1.8	0.7540 ± 1.4	0.8062 ± 1.7	0.8426 ± 1.4	0.8654 ± 1.5	0.8896 ± 1.6	0.9034 ± 0.7
R-UMLDA AdaboostM2 *k*-median	0.2950 ± 2.1	0.5060 ± 3.4	0.6292 ± 1.3	0.6994 ± 1.8	0.7580 ± 2.2	0.7970 ± 1.9	0.8318 ± 2.0	0.8606 ± 1.8	0.8964 ± 1.7	0.9100 ± 1.3

Table 4.5 (b) Testing CCR (Mean±Std%): image data with 5 training samples per class

Change	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
R-UMLDA - A	0.1903 ± 0.8	0.2486 ± 3.6	**0.3350 ± 4.8**	**0.4408 ± 6.8**	**0.4896 ± 3.1**	**0.5874 ± 6.4**	**0.6608 ± 4.8**	**0.7092 ± 4.0**	**0.7344 ± 5.2**	**0.7761 ± 3.7**
R-UMLDA Bagging	**0.2018 ± 1.4**	**0.2531 ± 3.3**	0.3310 ± 4.9	0.4090 ± 5.5	0.4613 ± 2.2	0.5248 ± 3.9	0.6294 ± 4.0	0.6627 ± 5.1	0.6944 ± 6.1	0.7443 ± 4.6
R-UMLDA Bagging *k*-means	0.1995 ± 1.1	0.2442 ± 2.6	0.3056 ± 3.0	0.3629 ± 4.7	0.4558 ± 5.8	0.5505 ± 5.8	0.5661 ± 4.4	0.5707 ± 11.3	0.6925 ± 7.3	0.7073 ± 4.7

Table 4.5 (c) Testing CCR (Mean±Std%): image data with 10 training samples per class

Change	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
R-UMLDA - A	**0.2113 ± 1.6**	0.2868 ± 3.3	0.3824 ± 3.8	**0.5056 ± 3.3**	0.5745 ± 6.3	**0.6582 ± 2.3**	**0.7057 ± 4.5**	**0.7502 ± 3.3**	**0.7917 ± 3.4**	**0.8315 ± 3.0**
R-UMLDA Bagging	0.1978 ± 1.1	0.2814 ± 3.0	0.4016 ± 4.3	0.4849 ± 4.8	0.5497 ± 6.2	0.6046 ± 5.0	0.6599 ± 4.0	0.7106 ± 3.8	0.7635 ± 3.6	0.7778 ± 4.3
R-UMLDA Bagging *k*-means	0.2017 ± 1.8	**0.2936 ± 1.7**	**0.4259 ± 4.6**	0.5035 ± 4.5	**0.5934 ± 1.7**	0.6462 ± 3.6	0.6996 ± 5.6	0.7475 ± 3.8	0.7782 ± 3.4	0.7952 ± 3.6

Table 4.5 (d) Testing CCR (Mean±Std%): image data with 20 training samples per class

Change	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
R-UMLDA - A	$\mathbf{0.2137 \pm 1.7}$	0.3377 ± 2.7	0.4927 ± 3.0	$\mathbf{0.6003 \pm 2.3}$	$\mathbf{0.6647 \pm 1.7}$	0.7417 ± 2.0	0.7727 ± 1.3	0.8089 ± 1.8	0.8319 ± 2.7	$\mathbf{0.8686 \pm 1.6}$
R-UMLDA Bagging	0.2117 ± 1.6	$\mathbf{0.3507 \pm 2.1}$	0.4789 ± 4.4	0.5599 ± 4.5	0.6484 ± 2.3	0.7269 ± 2.0	0.7522 ± 2.0	0.7999 ± 2.8	0.8321 ± 2.2	0.8486 ± 2.8
R-UMLDA Bagging *k*-means	0.2043 ± 1.6	0.3489 ± 2.6	$\mathbf{0.5184 \pm 4.0}$	0.5890 ± 3.5	0.6489 ± 3.9	$\mathbf{0.7429 \pm 2.3}$	$\mathbf{0.7836 \pm 2.5}$	$\mathbf{0.8189 \pm 3.0}$	$\mathbf{0.8362 \pm 1.2}$	0.8662 ± 2.2

4.5 Application in Friction Stir Blind Riveting

Now, we apply the proposed method to the FSBR data studied in Chapter 3. Different from the *in-situ* NDE method we developed in Chapter 3, we will not use engineering knowledge or process settings in this case study. We will demonstrate the proposed method and compare its performance with the competitors used in the numerical study.

After pre-processing, the *in-situ* process signals are represented as a tensor object $\mathcal{A} \in \mathbb{R}^{600\times2\times18}$, where 600 is the pre-processed signal length, 2 represents two sensors and 18 is the number of FSBR samples with quality information from tensile tests. Since the training sample size is limited and the sample size from each class is uneven, we modeled the real process signals with piecewise cubic splines and then generated surrogate signals to enlarge the training sample size. After adding surrogate signals to the real data, we have 7 dataset partitions: $\{8, 10, 12, 14, 16, 18, 20\}$ training samples in each quality group.

The low- and high-quality groups from Chapter 3 will still be used as class label in this study. The quality evaluation performance from our proposed method is compared with R-UMLDA-A and R-UMLDA- EL. Since the Adaboost method cannot provide a better result than random guess, we only consider Bagging in this case study. The quality

evaluation result from the optimal $\boldsymbol{\gamma}$ subinterval is shown in Figure 4.9. Each point represents the average testing CCR from 10 replications. The green curve with upper triangle markers represent the average testing CCR from R-UMLDA Bagging with k-median diversity maximization method. This method outperforms other methods at training sample size 8, 12, 14 and 16. R-UMLDA Bagging with k-means diversity maximization method performs the best at training sample size 10, 18 and performs the same as R-UMLDA-A at training sample size 20. Therefore, our proposed method wins 6 out of the 7 cases. The average and standard deviation of testing CCR are listed in Table 4.6. The optimal interval for $\boldsymbol{\gamma}^*$ is also $[10^{-7}, 10^{-2}]$ for FSBR process.

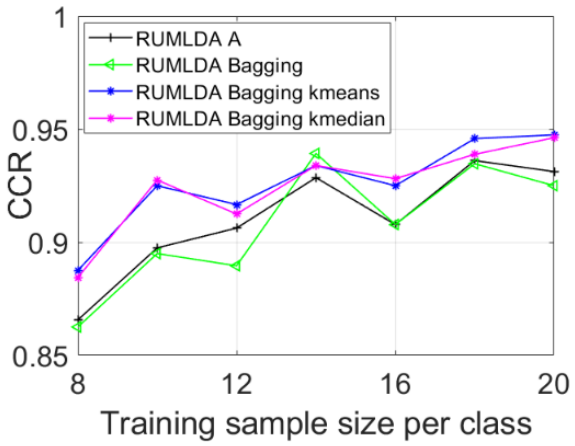


Figure 4.9 Case study result

Table 4.6 Testing CCR (Mean±Std%): FSBR data

Training Sample size per class	***8***	***10***	***12***	***14***	***16***	***18***	***20***
R-UMLDA - A	0.8750 ± 5.1	0.8750 ± 3.5	0.8875 ± 5.9	0.9179 ± 6.3	0.9219 ± 4.2	0.9278 ± 2.3	**0.9425 ± 2.9**
R-UMLDA Bagging	0.8688 ± 7.5	0.8600 ± 4.6	0.8833 ± 5.5	0.9107 ± 7.6	0.9094 ± 5.0	0.9222 ± 3.9	0.9325 ± 3.1
R-UMLDA Bagging *k*-means	0.8625 ± 5.7	**0.9050 ± 6.4**	0.9083 ± 4.7	0.9107 ± 6.4	**0.9250 ± 4.7**	**0.9417 ± 2.8**	**0.9425 ± 3.3**
R-UMLDA Bagging *k*-median	**0.8875 ± 4.9**	0.9013 ± 6.9	**0.9292 ± 4.0**	**0.9393 ± 4.1**	**0.9250 ± 4.2**	0.9333 ± 5.1	0.9325 ± 4.3

4.6 Conclusion

Tensor decomposition can effectively reduce the dimensionality of the input data and simplify the classification task. Ensemble learning can effectively stabilize the feature extraction procedure and strengthen the classification performance. In naïve integration, tensor decomposition and ensemble learning are performed sequentially but independently, without boosting their advantages in the integration. This chapter develops a novel integration of supervised tensor decomposition and ensemble learning to extract information from multi-sensor data for the purpose of enabling *in-situ* NDE. Supervised tensor decomposition is adopted to make full use of the available class information. The proposed integration method optimizes the feature extraction by considering ensemble learning performance; on the other hand, the method utilizes ensemble learning to determine the optimal regularization parameter for tensor decomposition.

The highlight of the proposed method is that it selects features that will perform the best in ensemble learning. Both Bagging and Adaboost are considered in this study. A diversity measure is defined to evaluate and infer how well the extracted features will perform in ensemble learning. According to the diversity metric, optimal features are selected such that additional diversity from the feature space can be introduced. Since the selected features should maximize the diversity among different feature groups and minimize the diversity in the same feature groups at the same time, clustering algorithms are suitable to help make the selection. Both k-means and k-median clustering methods are considered in our method. We also develop an adaptive approach to determine the optimal values for the regularization parameter in tensor decomposition.

The proposed method is evaluated in two numerical studies: one with multi-profile signal data and the other with image data. In both studies, the proposed method is evaluated in small and large training sample sizes, in multiple change patterns, and in various magnitudes of changes. These studies demonstrate the effectiveness of our method in extracting uncorrelated features from high-dimensional tensor data, monitoring various magnitudes of process changes, identifying the changes via classification, and handling the SSS problem. The proposed method is compared with a benchmark method and also naïve integration methods. Our method demonstrates to be superior to the competitors in both numerical studies and FSBR case study. The case study on FSBR signals also shows that our method is an effective enabler for the *in-situ* NDE of FSBR and other similar manufacturing processes.

For future research, it will be interesting to quantify the relationship between the regularization parameter γ and the eigenvalue $\lambda_i, i = 1, \cdots, I_l$. Friedman (1989) and Lu *et*

al. (2009) provided empirical studies on this topic, but no closed-form functions have been established. In Section 4.3.4, we consider the generalized ensemble learning error $\hat{e}$ as a convex function of the γ's subinterval $[a, b]$. This is an indirect heuristic approach to connect the regularized tensor decomposition with ensemble learning, without considering the eigenvalue λ. Establishing a closed-form function between λ and γ is likely to accelerate the search for optimal γ^*.

CHAPTER 5 CHARACTERIZATION OF TOOL WEAR IN MICRO-FRICTION STIR WELDING USING HIERARCHICAL SPATIO-TEMPORAL MODELS

Solid state joining processes such as friction stir welding (FSW) and micro friction stir welding (μFSW) usually require custom-made tools with specially designed profiles (Wang *et al*., 2018). The tool condition plays a significant role on process dynamics and part quality (Pfeifer and Wiegers, 2000; Haber *et al*., 2004; Shao *et al*., 2016). Tool wear is complex in the sense that it varies with the combination of tool-workpiece materials and joining conditions, which makes it quite challenging to detect. Effective modeling and monitoring of tool condition degradation can provide the technical basis for maintaining production efficiency and quality. Hence, tool condition monitoring in joining processes has gradually received attention (Shao *et al*., 2016).

Friction stir welding (FSW), patented at The Welding Institute (TWI) in the 1991 (Thomas *et al*., 1991), is a solid-state joining process wherein a non-consumable, cylindrical tool is rotated and traversed along a square butt weld joint using conventional milling techniques (Prado *et al*., 2001). μFSW is the adaptation of the FSW process to materials with thicknesses of 1000μm or less (Wang *et al*., 2018). Tool in μFSW has similar geometric design as tool in FSW (Scialpi *et al*., 2008). Since there is little published information available on the tool wear mechanism in μFSW, we consider the wear mechanism in FSW as an analogy for the tool wear mechanism in μFSW.

In FSW process, tool wear is the biggest factor governing tool useful life (Thompson, 2010). Most of the previous works characterized the FSW tool wear by dimensionally tracking the tool material loss given a length of weld (Thompson, 2010;

Weinberger *et al*., 2008). Further investigations divide tool wear into two mechanisms: abrasive wear and adhesive wear. The abrasive wear is defined as "abrasion caused by the displacement of material from a solid surface due to hard particles or protuberances sliding along the surface" and the adhesive wear is defined as "wear by transference of material from one surface to another during relative motion due to a process of solid-phase welding; particles that are removed from one surface are either permanently or temporarily attached to the other surface" (Weinberger *et al*., 2008). Sometimes, the abrasive wear and adhesive wear are concluded as abrasion and diffusion (Rai *et al*., 2011). Besides the material loss in abrasion and diffusion, reaction of the tool material with its environment also contributes to tool wear (Rai *et al*., 2011).

The FSW tool usually consists of a shoulder part and a pin (probe) part. Functions of the shoulder part and pin part can be described as follows: During welding, the shoulder is pressed against the surface of the welding material while the pin is forced between the shoulder and the welding material. Rotation of the tool generates frictional heat to soften and plastically deform the welding material. The softened material flows behind the rotating tool and forms a solid state weld (Scialpi *et al*., 2008). Figure 5.1 is a schematic illustration of an FSW butt joint that shows the tool shoulder and the tool pin (probe). Compared to the shoulder part, the tool pin suffers much more severe wear and hence, the tool failure almost always occurs on the pin part. The wear in the pin part is represented as the downward tool material loss. The material loss evolution for an FSW tool with threaded cylindrical pin is shown in Figure 5.2. The top three images display the evolution with a tool with travel speed of 3mm/s and the bottom three images display the evolution with a tool with travel speed of 9mm/s. The number at the bottom-right

corner of each image is the welding length in meters. It can be noticed that with a travel speed of 9mm/s, the tool loses material evenly along the cylindrical direction, which

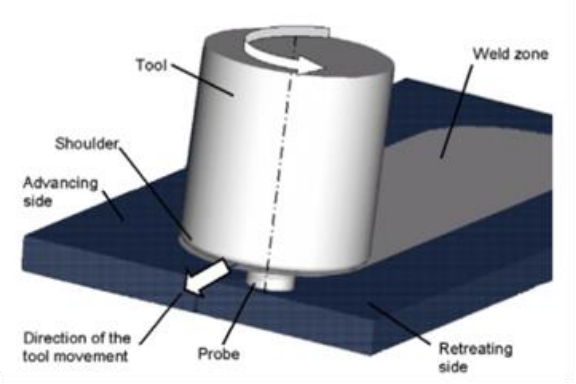


Figure 5.1 A schematic illustration of FSW butt-joint, adopted from Scialpi *et al*. (2008)

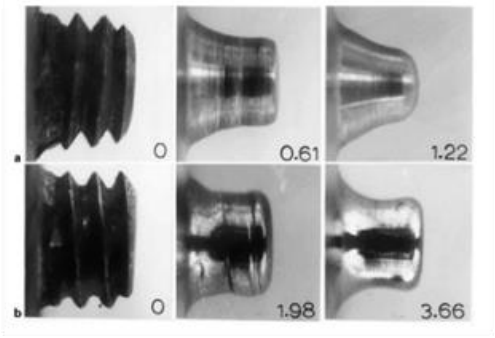


Figure 5.2 Tool pin downward material loss, adopted from Scialpi *et al*. (2008)

implies abrasion wear; with a travel speed of 3mm/s, the material loss spreads along the radius direction, which implies diffusion wear. The details on the tool pin wear mechanisms can be referred to Rai *et al*. (2011). Therefore, tool wear in μFSW can be characterized by the changes in the profile of the tool pin.

Compared to a fresh tool, a worn tool has a different pin geometry and accordingly, the associated in-process dynamics. Past research by several groups have developed physical models based on process mechanics (e.g., computational fluid dynamic method) to correlate tool condition with process information such as material flow and temperature distribution during welding for FSW (Su *et al*., 2015; Chen *et al*., 2018; Dörfler, 2008). However, the accuracy of physical models has been shown to be limited (Dörfler, 2008) due to the assumptions or simplifications made for model derivation.

Although *in-situ* tool inspection is usually difficult to perform, there are many indirect inspection approaches to obtain the geometry of the tool. For example, Shao *et al*.

(2016) made an impression of the tool on a weld coupon in ultrasonic metal welding. They measured deformation of the coupon under a microscope and then the inverse of the coupon image was created as a surrogate of the tool image. With this "impression method", the tool doesn't need to be disassembled from the welder, making the tool inspection process relatively easy and fast. When direct inspection is allowed, the tool can be directly measured under a microscope to obtain the high-resolution tool surface information (Kurada and Bradley, 1997; Castejón *et al*., 2007). Figure 5.3 shows the high-resolution tool surface images obtained from the offline inspection of a μFSW tool after 400 joints via a Keyence VR-3100 microscope. The stepped shape of this rotating tool can be easily seen in this figure. We also notice that the shoulder and pin do not have sharp edges, indicating wear. Figure 5.3 (b) displays the measurements of the height of the tool pin surface. The height value is the difference from the surface to a reference plane, shown as the baseline in Figure 5.3 (b). The geometric features of the tool, e.g., step height, step diameter, chamfer corner, etc., are measured in Figure 5.3 (c). The Keyence microscope in VR-3000 series can automatically discretize the measured tool pin surface profile into coordinated grids and each grid point stores the information of the pin profile height (μm) as well as the projection positions on the reference plane. Hence, the tool pin surface image, like Figure 5.3 (b), can be considered as functional data with spatial dependence. Collected over time, the sequence of tool pin surface images will provide rich information about the tool wear with both space and time information.

Recent studies on tool condition monitoring try to harness data such as tool surface images using machine learning methods. However, most of the existing studies on tool condition monitoring for FSW-type joining processes either ignore the temporal

correlation or the spatial dependence. On one hand, existing studies usually treat tool wear as a state (fresh vs. worn) rather than a process that is dynamically involving in time (Biswas *et al*., 2012; Chen *et al*., 2018). On the other hand, existing studies usually characterize tool wear by a single or a few synthetic indices, defined in a handcrafted way, such as the diameter of the tool pin (Kumar and Raju, 2012), the proportion of the remaining pin length to the designed length after a usage period (Rai *et al*., 2011) and so on. Tool condition monitoring that considers both spatial and temporal dependences in the tool surface images are needed for FSW-type joining processes.

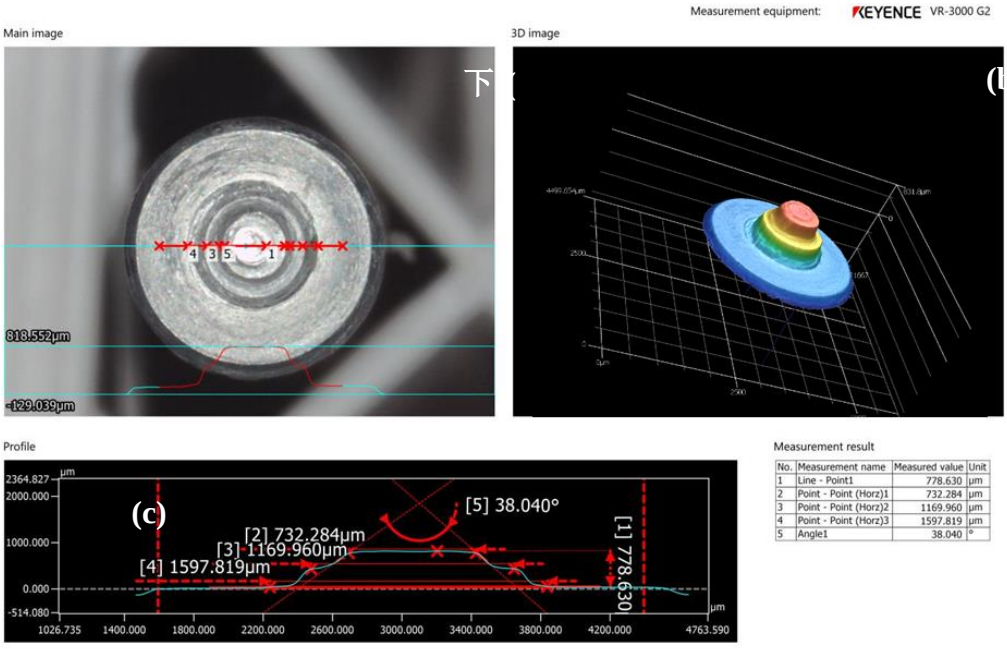


Figure 5.3 Inspected μFSW joining tool: (a) top view of the tool pin surface; (b) tool pin surface profile; (c) tool pin profile with extracted geometric features

In this chapter, we present a hybrid spatio-temporal model to characterize the dynamic tool wear propagation in micro friction stir welding. The proposed hybrid model is developed in a hierarchical Bayesian structure with the first level being a data-driven regression model for the high-resolution tool pin profile images and the second level

being a physics-based advection-diffusion model for the latent (unobserved) welding temperature distribution. Discrete Kalman filter is utilized to estimate (1) the mean of the posterior distribution of the state variable and (2) the error between the measured tool surface image and the mean of the fitted posterior distribution of the tool pin profile. To monitor tool wear progression, we define monitoring statistics from the two estimations using regularized Mahalanobis distance. Thresholds are determined to detect abnormal tool wear progression patterns. To demonstrate the effectiveness of the proposed method, we simulate three abnormal tool wear progression patterns and evaluate the performance of the proposed method in detection timeliness and accuracy.

5.1 Method Overview

The flowchart of the proposed method in Chapter 5 is shown in Figure 5.4. There are three major steps in the proposed method: (1) building the hierarchical spatio-temporal model, (2) fitting the hierarchical model to estimate the posterior distribution of the state variable and the tool pin profile via Kalman filter, and (3) monitoring tool wear progression and detecting abnormal patterns.

Tool pin profile images are taken at discrete time steps with equal time intervals using a high-resolution microscope. The image at the nth step is denoted as $\boldsymbol{Y}(n) \in \mathbb{R}^{I_1 \times I_2}$, where $I_1 \times I_2$ describes the coordinate grids of the measured tool image along horizontal and vertical directions; $n = 2, \cdots, N$ represents tool wear progression with termination step N.

In Step 1 of the proposed method, the tool pin profile image $\boldsymbol{Y}(n)$ is vectorized to $\boldsymbol{y}(n) \in \mathbb{R}^{I_1 I_2 \times 1}$. The latent (unobserved) welding temperature distribution is denoted as $\boldsymbol{c}(n) \in \mathbb{R}^{I_1 I_2 \times 1}$. $\boldsymbol{y}(n)$ is modeled in the first level of the hierarchical spatio-temporal

model as a linear combination of $\boldsymbol{c}(n)$ and an error term $\boldsymbol{\varepsilon}(n)$. The dynamic property of $\boldsymbol{c}(t)$ is characterized in the second level as a physics-based advection-diffusion model using partial differential equation (PDE).

Next, the hierarchical spatio-temporal model model is fitted in Step 2 to obtain the step-wise posterior distribution of the state variable $\hat{\boldsymbol{c}}[n|n]$ as well as the error $\hat{\boldsymbol{e}}[n]$ between the fitted posterior distribution of the tool pin profile $\hat{\boldsymbol{y}}[n|n]$ and tool surface measurement $\boldsymbol{y}(n)$. Step 2 includes three sub-steps: (1) Convert the advection-diffusion model into a linear state space model via finite difference approximation and so the proposed hierarchical spatio-temporal model is converted to a dynamic linear system; (2) Estimating all necessary parameters for the dynamic linear system; (3) Fitting the dynamic linear system via discrete Kalman filter.

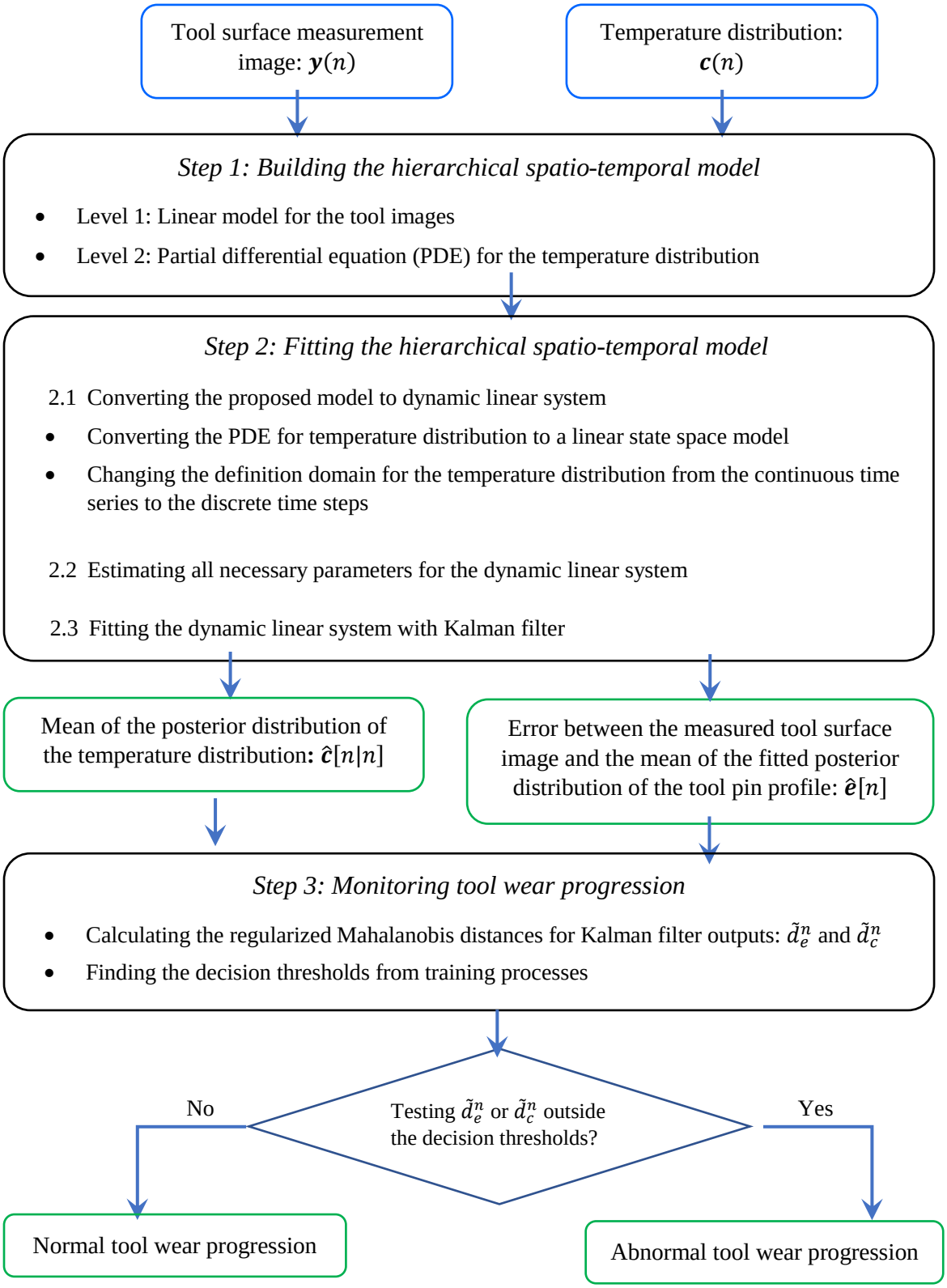


Figure 5.4 Flowchart of the proposed method

Tool wear progression is monitored in Step 3. Monitoring statistics from the two estimations using regularized Mahalanobis distance is defined first and then thresholds are determined to detect abnormal tool wear progression patterns. Three different abnormal tool wear progression patterns representing different tool wear mechanisms are simulated in numerical study. Numerical study performance demonstrates the effectiveness of the proposed method.

5.2 Hierarchical Spatial-Temporal Model for Tool Wear Progression

The hierarchical spatio-temporal modeling method is a Bayesian inference based modeling strategy and is widely used in dynamic processes with spatial correlations.

5.2.1 Bayesian Inference

In spatio-temporal statistics, a set of observations in space and time can be regarded as a sample collected from a spatio-temporal process (Wikle *et al*., 1998). Without loss of generality, we consider the discrete time step in this section. Following the notations in 5.1, at any time step $n, n = 2, \cdots, N$, the measured tool surface image $\boldsymbol{y}(n)$ is a sample drawn from the μFSW tool wear progression process, which is denoted as $\boldsymbol{c}(n)$. This sampling can be represented as conditional probability $p\big(\boldsymbol{y}(n)|\boldsymbol{c}(n)\big)$. Let $\pi(\boldsymbol{c})$ denote the prior distribution of the process. The posterior distribution of the tool wear process is:

$$p\big(\boldsymbol{c}(n)\,|\boldsymbol{y}(n)\big) \propto p\big(\boldsymbol{y}(n)\,|\boldsymbol{c}(n)\big) \times \pi(\boldsymbol{c}). \tag{5.1}$$

To characterize a spatio-temporal process in the hierarchical Bayesian structure, a joint distribution is derived first to model the data, process, and parameters in a spatio-temporal process together and then the joint distribution is hierarchically decomposed into a series of conditional probabilities (Berliner *et al*., 2000). In Bayesian inference, all

the decomposed terms for a spatio-temporal process can be treated as individual models. The hierarchical decomposition is written as:

$$\pi(\text{data}, \text{process}, \text{parameter})$$
$$\propto \pi(\text{data}|\text{process}, \text{parameter})\pi(\text{process}|\text{parameter})\pi(\text{parameter}). \quad (5.2)$$

The three models in the right hand side of Eq. (5.2) are (1) the data distribution model $\pi(\text{data}|\text{process}, \text{parameter})$, which models the tool surface measurement image, (2) the prior process model $\pi(\text{process}|\text{parameter})$, which describes the temperature distribution during welding, and (3) the prior on parameters $\pi(\text{parameter})$, which refers to all unknown parameters for the measurement data and tool wear process. These three models form a hierarchical structure. The data distribution model is usually set in the first level as a linear combination of the process and an error term. In tool wear characterization, the tool pin profile $\boldsymbol{y}(n)$ is a linear combination of the temperature distribution $\boldsymbol{c}(n)$ and an error term $\boldsymbol{\varepsilon}(n)$. The error term is usually further divided into a spatially correlated term and a spatially uncorrelated term. We assume $\boldsymbol{\varepsilon}(n)$ is spatially correlated but temporally independent, in order to capture all possible uncertainties during tool profile modeling. The temperature distribution in μFSW is modeled in the second level to characterize its dynamics, based on the physics of μFSW (Hirasawa *et al*., 2010). The parameters for the first and second level models are further divided into a stochastic set that should be estimated via statistical inference methods and a deterministic set that can be defined and obtained via manufacturing setting and computational fluid dynamic methods.

5.2.2 Linear Model for Tool Pin Profile

In the first level of the hierarchical spatio-temporal model, the tool pin profile measurement is modeled as a linear combination of the tool wear process and an error term that captures all uncertainties in modeling. Since the temperature distribution in the tool wear process is defined in continuous time series, the data distribution model is also defined in continuous time series first:

$$\boldsymbol{y}(t) = \boldsymbol{H}\boldsymbol{c}(t) + \boldsymbol{\varepsilon}(t), \tag{5.3}$$

where $\boldsymbol{c}(t) \in \mathbb{R}^{I_1 I_2 \times 1}$ is the temperature distribution in the tool wear process in time series, $t \in [0, T)$, and T is the tool wear termination time, i.e., useful life of the tool. $\boldsymbol{y}(t) \in \mathbb{R}^{I_1 I_2 \times 1}$ is the tool pin profile at time t. It is noted that $\boldsymbol{y}(t)$ cannot be measured in real-time in μFSW, due to process condition and complexity. The tool pin profile in continuous time series can only be estimated from the proposed hierarchical model. $\boldsymbol{H} \in \mathbb{R}^{I_1 I_2 \times I_1 I_2}$ is the projection matrix to connect the process to the tool surface measurement. Although it is widely acknowledged that there is a strong correlation between the welding temperature distribution and the tool pin profile, a closed-form equation that can connect these two variables has not been derived yet. We add an error term $\boldsymbol{\varepsilon}(t)$ to compensate for the modeling uncertainties. In spatio-temporal models, $\boldsymbol{\varepsilon}(t)$ is assumed to follow a Gaussian distribution, $\boldsymbol{\varepsilon}(t) \sim Gau(0, \boldsymbol{Q_1})$, where $\boldsymbol{Q_1}$ is the covariance matrix.

In spatio-temporal processes, without loss of generality, we assume that $\boldsymbol{\varepsilon}(t)$ is temporally independent and that it can be characterized by a Matern format spatio-temporal covariance function. For any two different coordinate grid points in the tool measurement image, $i \neq j$, the covariance function of $\boldsymbol{\varepsilon}(t)$ can be expressed as:

$$\mathrm{Cov}\left[\boldsymbol{\varepsilon}\left(\boldsymbol{s}_{ix},\boldsymbol{s}_{iy},t\right),\boldsymbol{\varepsilon}\left(\boldsymbol{s}_{jx},\boldsymbol{s}_{jy},t'\right)\right]=\begin{cases}0, if\ t\neq t'\\ \sigma_{\varepsilon}^{2}\boldsymbol{\mathcal{C}}(h), if\ t=t'\end{cases}, \tag{5.4}$$

where $\boldsymbol{\mathcal{C}}(h)$ is the pure spatial correlation function of the two different locations $\left(\boldsymbol{s}_{ix},\boldsymbol{s}_{iy}\right)$ and $\left(\boldsymbol{s}_{jx},\boldsymbol{s}_{jy}\right)$, depending on the Euclidean distance $h=\left\|\left(\boldsymbol{s}_{ix},\boldsymbol{s}_{iy}\right)-\left(\boldsymbol{s}_{jx},\boldsymbol{s}_{jy}\right)\right\|$. $\boldsymbol{\varepsilon}(t)$ is assumed to be second-order stationary and isotropic. The spatial correlation function $\boldsymbol{\mathcal{C}}(h)$ is a Matern function given by:

$$\boldsymbol{\mathcal{C}}(h)=\frac{1}{\Gamma(\nu)2^{\nu-1}}(\kappa h)^{\nu}K_{\nu}(\kappa h), \tag{5.5}$$

where $K_{\nu}(\cdot)$ denotes the modified Bessel function of the second kind. ν is the degree of smoothness of the process and its value determines the mean square differentiability of the process, $\nu>0$. $\kappa>0$ is a scaling parameter. σ_{ε}^{2} in Eq. (5.4) is the marginal covariance. Detailed explanation of Eq. (5.5) is presented in Lindgren *et al*. (2011). Therefore, the independent parameters in the data distribution model for $\boldsymbol{y}(t)$ are $\boldsymbol{\theta}_1=\{\sigma_{\varepsilon}^{2},\nu,\kappa\}$. They are stochastic parameters and are usually estimated via Bayesian inference based statistical methods.

5.2.3 Partial Differential Equation for Welding Temperature Distribution

Since μFSW is a novel variant of FSW, its temperature distribution conduction equation is not available in literature. We adopt the unsteady thermal conduction equation for Friction Stir Spot Welding (FSSW) (Hirasawa *et al*., 2010) for μFSW in this chapter. FSSW is another variant of FSW. The unsteady thermal conduction equations in different variants of FSW are very similar, with minimal difference on parameter assumptions. Considering the temperature distribution along the horizontal and vertical dimensions only, the process variable can be expressed as:

$$\rho a\left(\frac{\partial \boldsymbol{c}(t)}{\partial t}+u\frac{\partial \boldsymbol{c}(t)}{\partial \boldsymbol{s}_x}+v\frac{\partial \boldsymbol{c}(t)}{\partial \boldsymbol{s}_y}\right)=\lambda\left(\frac{\partial^2 \boldsymbol{c}(t)}{\partial {\boldsymbol{s}_x}^2}+\frac{\partial^2 \boldsymbol{c}(t)}{\partial {\boldsymbol{s}_y}^2}\right)+q, \tag{5.6}$$

where $\boldsymbol{s}_x$ and $\boldsymbol{s}_y$ are the coordinates along the horizontal and vertical directions (m), respectively, u and v are flow velocity (m/s), ρ is the welding material density (kg/m^3), a is the heat capacity (J/(kg·K)), λ is the thermal conductivity (W/(m· K)), and q is the heat generation rate per unit volume (W/m^3)). $\boldsymbol{c}(t)$ is defined in Eq. (5.3) with unit (°C). After separating the partial derivatives according to the thermal conduct time and the thermal conduct area, we can convert the temperature distribution conduction equation in Eq. (5.6) into:

$$\frac{\partial \boldsymbol{c}(t)}{\partial t} = -\left\{\frac{\partial(u\boldsymbol{c}(t))}{\partial \boldsymbol{s}_x} + \frac{\partial(v\boldsymbol{c}(t))}{\partial \boldsymbol{s}_y}\right\} + \frac{\lambda}{\rho a}\left\{\frac{\partial^2 \boldsymbol{c}(t)}{\partial \boldsymbol{s}_x{}^2} + \frac{\partial^2 \boldsymbol{c}(t)}{\partial \boldsymbol{s}_y{}^2}\right\} + \frac{q}{\rho a}. \tag{5.7}$$

The parameters in the Eq. (5.7) are $\boldsymbol{\theta}_2 = \{u, \lambda, \rho, a, q\}$. They are welding process parameters and are usually determined by manufacturing settings and can be calculated through computational fluid dynamics methods. So far we have laid out the models for data distribution and process, as well as the parameters in the hierarchical model for μFSW tool wear progression. The next step is to estimate the posterior distributions for the Bayesian inference based hierarchical model.

5.3 Model Fitting via Discrete Kalman Filter

Bayesian inferences are widely adopted to fit the hierarchical spatio-temporal model and to estimate parameters. Existing work on this topic is reviewed in Chapter 2. However, our models in Eq. (5.3) and Eq. (5.7) cannot be directly handled by the standard Bayesian inference methods. The specific issues are as follows.

- The current hierarchical model is based on continuous time series while the tool pin profile images are taken in discrete time steps. There may be large errors if we fit the model in continuous time series with limited discrete observation data.

- The standard method for model fitting and parameter estimation in Bayesian inferences is using Monte Carlo methods. Although Monte Carlo methods can provide the posterior distribution for parameters with high accuracy, the computational complexity for a Monte Carlo method is usually huge and the variance of the posterior distribution tend to be either underestimated or overestimated (Rue *et al*., 2009) and therefore, the applicability of Bayesian inference could be undermined.

A unique feature about the partial differential equation in Eq. (5.7) is that it has the advection-diffusion format. Many methods, e.g., finite differences, Runge-Kutta, explicit, implicit, etc. can discretize a partial differential equation in advection-diffusion format (Wikle and Hooten, 2010). After discretization, the hierarchical spatio-temporal model can be converted to a dynamic linear system and the process variable as well as the data distribution will be easily estimated via a discrete Kalman filter.

5.3.1 Model Discretization Approximation

The process model in Eq. (5.7) is a partial differential equation in advection-diffusion format. In Eq. (5.7), the term $\frac{\partial(u\boldsymbol{c}(t))}{\partial \boldsymbol{s_x}} + \frac{\partial(v\boldsymbol{c}(t))}{\partial \boldsymbol{s_y}}$ is the first-order gradient of the temperature distribution and it is the advection term. Assuming that the welding flow velocity is the same towards different directions, $u = v$, then u is the advection parameter. The advection term describes that the welding temperature distributes along the horizontal and vertical directions with the movement of the welding tool. The term $\frac{\partial^2 \boldsymbol{c}(t)}{\partial \boldsymbol{s_x}^2} + \frac{\partial^2 \boldsymbol{c}(t)}{\partial \boldsymbol{s_y}^2}$ is the second-order gradient of the temperature distribution and it is the

diffusion term. $\frac{\lambda}{\rho a}$ is the diffusion parameter. The diffusion term denotes that the welding temperature spreads out over a larger and larger welding area along time series.

The forward-time, central-space (FTCS) finite difference method is used to approximate the partial differential equation in advection-diffusion format with a linear model. The partial derivatives approximations are:

$$\begin{aligned} \frac{\partial \boldsymbol{c}(t)}{\partial t} &\approx \frac{\boldsymbol{c}(\boldsymbol{s}_x,\boldsymbol{s}_y,t)-\boldsymbol{c}(\boldsymbol{s}_x,\boldsymbol{s}_y,t-\Delta t)}{\Delta t} \\ \frac{\partial \boldsymbol{c}(t)}{\partial \boldsymbol{s}_x} &\approx \frac{\boldsymbol{c}(\boldsymbol{s}_x+\Delta \boldsymbol{s}_x,\boldsymbol{s}_y,t)-\boldsymbol{c}(\boldsymbol{s}_x-\Delta \boldsymbol{s}_x,\boldsymbol{s}_y,t)}{2\Delta \boldsymbol{s}_x} \\ \frac{\partial^2 \boldsymbol{c}(t)}{\partial {\boldsymbol{s}_x}^2} &\approx \frac{\boldsymbol{c}(\boldsymbol{s}_x+\Delta \boldsymbol{s}_x,\boldsymbol{s}_y,t)+\boldsymbol{c}(\boldsymbol{s}_x-\Delta \boldsymbol{s}_x,\boldsymbol{s}_y,t)-2\boldsymbol{c}(\boldsymbol{s}_x,\boldsymbol{s}_y,t)}{(\Delta \boldsymbol{s}_x)^2} \end{aligned}, \tag{5.8}$$

where $\Delta \boldsymbol{s}_x$, $\Delta \boldsymbol{s}_y$, and Δt denote the spatial and temporal spacings. Using the discretization method, the partial differential equation for the temperature distribution can be written as:

$$\boldsymbol{c}(\boldsymbol{s}_x,\boldsymbol{s}_y,t+\Delta t) = \phi^1_{\boldsymbol{s}_x\boldsymbol{s}_y}\boldsymbol{c}(\boldsymbol{s}_x,\boldsymbol{s}_y,t) + \phi^2_{\boldsymbol{s}_x\boldsymbol{s}_y}\boldsymbol{c}(\boldsymbol{s}_x+\Delta \boldsymbol{s}_x,\boldsymbol{s}_y,t) + \phi^3_{\boldsymbol{s}_x\boldsymbol{s}_y}\boldsymbol{c}(\boldsymbol{s}_x - \Delta \boldsymbol{s}_x,\boldsymbol{s}_y,t) + \phi^4_{\boldsymbol{s}_x\boldsymbol{s}_y}\boldsymbol{c}(\boldsymbol{s}_x,\boldsymbol{s}_y+\Delta \boldsymbol{s}_y,t) + \phi^5_{\boldsymbol{s}_x\boldsymbol{s}_y}\boldsymbol{c}(\boldsymbol{s}_x,\boldsymbol{s}_y-\Delta \boldsymbol{s}_y,t) + \frac{q}{\rho a}, \tag{5.9}$$

where $\phi^1_{\boldsymbol{s}_x\boldsymbol{s}_y} = 1 - \left(\frac{\lambda}{\rho c}\frac{2}{(\Delta s_x)^2} + \frac{\lambda}{\rho c}\frac{2}{(\Delta s_y)^2}\right)\Delta t$, $\phi^2_{\boldsymbol{s}_x\boldsymbol{s}_y} = \phi^4_{\boldsymbol{s}_x\boldsymbol{s}_y} = \left(\frac{\lambda}{\rho c}\frac{1}{(\Delta s_x)^2} - \frac{u}{2\Delta s_y}\right)\Delta t$, and $\phi^3_{\boldsymbol{s}_x\boldsymbol{s}_y} = \phi^5_{\boldsymbol{s}_x\boldsymbol{s}_y} = \left(\frac{\lambda}{\rho c}\frac{1}{(\Delta s_x)^2} + \frac{u}{2\Delta s_y}\right)\Delta t$ are coefficients of the spatial location $(\boldsymbol{s}_x,\boldsymbol{s}_y)$ in the transition matrix $\boldsymbol{\Phi}(t)$ that connects the temperature distribution in any consecutive two time steps. The welding area is assumed to be a two-dimensional coordinate grids with size $I_1 \times I_2$, $\boldsymbol{s}_x = 1,\cdots,I_1$ and $\boldsymbol{s}_y = 1,\cdots,I_2$.

To derive $\boldsymbol{\Phi}(t)$, we order the grid points column by column and let $\boldsymbol{c}(t) = \left(c(1,1,t),\cdots,c(1,I_1,t),c(2,1,t),\cdots,c(2,I_1,t),\cdots,c(I_2,I_1,t)\right)'$ as an $(I_1I_2)\times 1$ vector of the temperature distribution at time t. Assume $\Delta \boldsymbol{s}_x = \Delta \boldsymbol{s}_y = \Delta t = 1$. The coefficients in

$\boldsymbol{\Phi}(t)$ will be $\phi^1_{s_x s_y} = 1 - \frac{4\lambda}{\rho a}$, $\phi^2_{s_x s_y} = \phi^4_{s_x s_y} = \frac{\lambda}{\rho a} - \frac{u}{2}$, and $\phi^3_{s_x s_y} = \phi^5_{s_x s_y} = \frac{\lambda}{\rho a} + \frac{u}{2}$. Since t is not expressed in these coefficients, $\boldsymbol{\Phi}(t)$ is time invariant and can be simplified to $\boldsymbol{\Phi}$. Take the grid boundary into consideration, the transition matrix is derived as:

$$\boldsymbol{\Phi} = \begin{Bmatrix}
\phi^1 & \phi^5 & 0 & \cdots & 0 & \phi^2 & & & & & & & & & & \\
\phi^4 & \phi^1 & \phi^5 & 0 & \cdots & 0 & \phi^2 & & & & & & & & & \\
0 & & & & & & & & & & & & & & & \\
\vdots & \ddots & \ddots & & & \ddots & \ddots & \ddots & & & & & & & & \\
0 & & & & & & & & & & & & & & & \\
\phi^3 & 0 & \cdots & 0 & \phi^4 & \phi^1 & \phi^5 & 0 & \cdots & 0 & \phi^2 & & & & & \\
 & & & & & \ddots & \ddots & \ddots & \ddots & & & & & & & \\
 & & & & & \phi^3 & 0 & \cdots & 0 & \phi^4 & \phi^1 & \phi^5 & 0 & \cdots & 0 & \phi^2 \\
 & & & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & & & & 0 \\
 & & & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & & & & \vdots \\
 & & & & & & \ddots & \ddots & \ddots & \ddots & \ddots & & & & & 0 \\
 & & & & & & & & & \phi^3 & 0 & \cdots & 0 & \phi^4 & \phi^1 & \phi^5 \\
 & & & & & & & & & & \phi^3 & 0 & \cdots & 0 & \phi^4 & \phi^1
\end{Bmatrix}_{I_1 I_2 \times I_1 I_2}. \tag{5.10}$$

Let (i, j) denote the coordinates of a grid point in the welding area. Using an interior point as an example, the transition matrix $\boldsymbol{\Phi}$ can be derived as follows. The state variable can jump from (i, j) to $(i+1, j)$, $(i-1, j)$, $(i, j+1)$, and $(i, j-1)$ with the transition parameters of ϕ^2, ϕ^3, ϕ^4, and ϕ^5, respectively. After ordering the grid points column by column, the state variable can move from (i, j) to its vertical neighbors $(i, j+1)$ and $(i, j-1)$ in one spatial step. In $\boldsymbol{\Phi}$, this is represented as the adjacent coefficients ϕ^4 and ϕ^5 to ϕ^1, where ϕ^1 means that the state variable doesn't move. However, after ordering, the state variable should spend $I_2 - 1$ spatial steps from (i, j) to its horizontal neighbors $(i+1, j)$ and $(i-1, j)$. In $\boldsymbol{\Phi}$**,** this is represented as $I_2 - 1$ zeros between ϕ^5 and ϕ^2 or between ϕ^4 and ϕ^3. The derivation is the same for the grid points on the boundaries or at the corners of the welding area. Zeros in $\boldsymbol{\Phi}$ means no possible

transition between the two gird points. Based on $\boldsymbol{\Phi}$, Eq. (5.7) is therefore approximated as:

$$\boldsymbol{c}(\boldsymbol{s}_x, \boldsymbol{s}_y, n+1) = \boldsymbol{\Phi}\boldsymbol{c}(\boldsymbol{s}_x, \boldsymbol{s}_y, n) + \boldsymbol{\omega}(\boldsymbol{s}_x, \boldsymbol{s}_y, n), \tag{5.11}$$

where $\boldsymbol{\omega}(\boldsymbol{s}_x, \boldsymbol{s}_y, n)$ is the model error term. Without loss of generality, we assume $\boldsymbol{\omega}(\boldsymbol{s}_x, \boldsymbol{s}_y, n) \sim Gau\left(\frac{q}{\rho a}\mathbf{1}, \boldsymbol{Q_2}\right)$ to account for the various sources of uncertainty during the finite difference approximation, $\mathbf{1} \in \mathbb{R}^{I_1 I_2 \times 1}$ is a vector of ones. Hence, $\boldsymbol{\omega}(\boldsymbol{s}_x, \boldsymbol{s}_y, n)$ has the same covariance structure as $\boldsymbol{\varepsilon}(\boldsymbol{s}_x, \boldsymbol{s}_y, n)$. $\boldsymbol{\omega}(\boldsymbol{s}_x, \boldsymbol{s}_y, n)$ is also time invariant and the finite differential approximation won't affect the temporal independence of $\boldsymbol{\omega}(\boldsymbol{s}_x, \boldsymbol{s}_y, n)$. Since the tool pin profile height and the temperature distribution change in the same welding area, we can assume that the parameters in $\boldsymbol{\omega}(\boldsymbol{s}_x, \boldsymbol{s}_y, n)$ are the same as the parameters in $\boldsymbol{\varepsilon}(\boldsymbol{s}_x, \boldsymbol{s}_y, n)$, except the variance parameter σ_ω^2. Therefore, after the finite difference approximation, we have another parameter $\boldsymbol{\theta}_3 = \{\sigma_\omega^2\}$. The hierarchical spatio-temporal model is consequently converted to the dynamic linear system:

$$\begin{aligned} \boldsymbol{y}(n) &= \boldsymbol{H}\boldsymbol{c}(n) + \boldsymbol{\varepsilon}(n) \\ \boldsymbol{c}(n+1) &= \boldsymbol{\Phi}\boldsymbol{c}(n) + \boldsymbol{\omega}(n) \end{aligned}. \tag{5.12}$$

In Eq. (5.12)'s dynamic linear system, both the data distribution $\boldsymbol{y}(n)$ and the process variable $\boldsymbol{c}(n)$ which are previously defined in continuous time domain have been discretized into discrete time steps. Discrete Kalman filter can be adopted to estimate the posterior distributions for these two variables.

Discrete Kalman filter is a mature technique to fit the dynamic linear system (Petris *et al*., 2009). Assuming the initial state of the process variable $\boldsymbol{c}(1) = \boldsymbol{\omega}(1)$, this dynamic linear model is uniquely determined by $\{\boldsymbol{H}, \boldsymbol{\Phi}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3\}$. Given values of $\{\boldsymbol{H}, \boldsymbol{\Phi}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3\}$, the posterior distribution of the state variable $\hat{\boldsymbol{c}}[n|n]$ and the fitting

error $\hat{\boldsymbol{e}}(n)$ for all $n = 1, \cdots, N$ can be obtained via discrete Kalman filter estimation. The fitting error $\hat{\boldsymbol{e}}(n)$ is the difference between the mean of the fitted posterior distribution of the tool pin profile and the tool pin profile measurement, calculated as:

$$\hat{\boldsymbol{e}}(n) = \hat{\boldsymbol{y}}[n|n] - \boldsymbol{y}(n). \tag{5.13}$$

Among all the parameters, elements in $\boldsymbol{\theta}_2$ are physics-based parameters. The specific values of $\boldsymbol{\theta}_2$ can be found in FSSW literatures. The $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_3$ parameters should be determined before the Kalman filter estimation. Note that $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_3$ are pure data-driven parameters for the Matern format covariance functions to capture the spatial structure of the welding area and they are temporally independent. This inspires us to develop another hierarchical spatio-temporal model for the tool profile images to determine $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_3$.

5.3.2 Parameter Estimation

In order to determine $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_3$ for the two Matern format covariance functions in the proposed model, we propose a pure data-driven hierarchical spatio-temporal model for the tool profile images. The parameters are denoted as $\boldsymbol{\theta}' = \{\sigma_\varepsilon^2, \sigma_\omega^2, \nu, \kappa\}$. Following the same hierarchical modeling framework as 5.2.2, we assume the data distribution is a linear combination of the spatial coordinates for the welding area and the error term. We further divide the error term into a spatial-temporal random effect to capture all uncertainties in time and space domains and a measurement error that is spatially uncorrelated and temporally independent. The data distribution model can be expressed as:

$$\boldsymbol{y}(t) = \boldsymbol{s}_x\beta_1 + \boldsymbol{s}_y\beta_2 + \boldsymbol{\xi}\left(\boldsymbol{s}_x, \boldsymbol{s}_y, t\right) + \boldsymbol{e}, \tag{5.14}$$

where $\boldsymbol{y}(t) \in \mathbb{R}^{I_1 I_2 \times 1}$ is the same tool pin profile at time t as in Eq. (5.3); $(\boldsymbol{s}_x, \boldsymbol{s}_y)$ is the location coordinates. Both $\boldsymbol{s}_x$ and $\boldsymbol{s}_y$ have the same dimension as $\boldsymbol{y}(t)$; $\boldsymbol{e}$ is the measurement error and spatially uncorrelated, $\boldsymbol{e} \sim MVN(\boldsymbol{0}, \sigma_\varepsilon^2 \boldsymbol{I})$, where $\boldsymbol{I} \in \mathbb{R}^{I_1 I_2 \times I_1 I_2}$ is the identity matrix; $\boldsymbol{\xi}(\boldsymbol{s}_x, \boldsymbol{s}_y, t)$ is the spatial-temporal random effect and the dynamics of $\boldsymbol{\xi}(\boldsymbol{s}_x, \boldsymbol{s}_y, t)$ is characterized by a process model in the second level. The process model is an AR(1) model written as:

$$\boldsymbol{\xi}(\boldsymbol{s}_x, \boldsymbol{s}_y, t+1) = \alpha \boldsymbol{\xi}(\boldsymbol{s}_x, \boldsymbol{s}_y, t) + \boldsymbol{\omega}'(\boldsymbol{s}_x, \boldsymbol{s}_y, t), \tag{5.15}$$

where α is a scalar and $0 < \alpha < 1$; $\boldsymbol{\omega}'(\boldsymbol{s}_x, \boldsymbol{s}_y, t)$ is the error term in the AR(1) model. We assume that $\boldsymbol{\omega}'(\boldsymbol{s}_x, \boldsymbol{s}_y, t)$ has the same distribution as $\boldsymbol{\omega}(\boldsymbol{s}_x, \boldsymbol{s}_y, t)$ in our proposed model and therefore $\boldsymbol{\omega}'(\boldsymbol{s}_x, \boldsymbol{s}_y, t) \sim Gau(0, \boldsymbol{Q}_{\boldsymbol{\omega}'})$. $\boldsymbol{Q}_{\boldsymbol{\omega}'}$ has the same structure as the covariance matrix $\boldsymbol{Q}_2$, $\boldsymbol{Q}_{\boldsymbol{\omega}'} = \sigma_\omega^2 \boldsymbol{\mathcal{C}}(h)$. The initial state of $\boldsymbol{\xi}(\boldsymbol{s}_x, \boldsymbol{s}_y, t)$ is given as $\boldsymbol{\xi}(\boldsymbol{s}_x, \boldsymbol{s}_y, 1) \sim N\left(0, {\boldsymbol{\mathcal{C}}(h)}/{(1-\alpha^2)}\right)$. Thus, in the newly designed hierarchical spatio-temporal model, all parameters are expressed in $\boldsymbol{\theta}' = \{\sigma_\varepsilon^2, \sigma_\omega^2, \nu, \kappa, \alpha, \beta_1, \beta_2\}$.

Monte Carlo method is one of the viable methods that can be adopted to fit this new hierarchical model and estimate the posterior distributions for all the parameters in this model. However, considering the drawbacks of Monte Carlo method, we propose to adopt the recently developed integrated nested Laplace approximation (INLA) method (Rue *et al*., 2009) with a stochastic partial differential equation (SPDE) (Lindgren *et al*., 2011) instead of Monte Carlo to obtain more reliable estimations of the posterior distributions.

It has been theoretically proven that the Matern format covariance function for a spatio-temporal process is the stationary solution of a stochastic partial differential equation (SPDE) (Whittle, 1954). SPDE is a model simplification technique in spatio-temporal statistics (Lindgren *et al*., 2011) to convert a Gaussian field into a Gaussian Markov random field (GMRF). The covariance matrix is converted from a dense matrix in the Gaussian field to a sparse matrix in GMRF. The spatial locations in a GMRF is only correlated with elements in its neighborhood. Details of the Matern format covariance function and SPDE can be found in Lindgren *et al.*, (2011). After SPDE, the error term $\boldsymbol{\omega}'(\boldsymbol{s}_x, \boldsymbol{s}_y, t)$ in AR(1) of the new hierarchical model is converted to $\boldsymbol{\omega}_2{}'(\boldsymbol{s}_x, \boldsymbol{s}_y, t) \sim GMRF(0, \boldsymbol{P}_{\boldsymbol{\omega}'})$, where $\boldsymbol{P}_{\boldsymbol{\omega}'}$ is the sparse covariance matrix with Markovian property. Given vague priors, the INLA method is adopted to estimate the posterior distributions of all parameters, denoted as $p(\boldsymbol{\theta}'|\boldsymbol{y}(t))$.

5.3.3 Model Fitting via Discrete Kalman Filter

We choose the mean of the posterior distribution of $p(\sigma_\varepsilon|\boldsymbol{y}(t)), p(\sigma_\omega|\boldsymbol{y}(t)), p(\nu|\boldsymbol{y}(t))$, and $p(\kappa|\boldsymbol{y}(t))$ as the parameters in $\boldsymbol{\theta_1}$ and $\boldsymbol{\theta_3}$. Values from Hirasawa *et al.*, (2005) are adopted for parameter $\boldsymbol{\theta}_2 = \{u, \lambda, \rho, a, q\}$ and the transition matrix $\boldsymbol{\Phi}$. Since no close-form model can be found to describe the relationship between μFSW tool pin profile and the temperature distribution, we set the transition matrix $\boldsymbol{H}$ as $\boldsymbol{I}$. In this way, we have determined all values of $\{\boldsymbol{H}, \boldsymbol{\Phi}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3\}$. Through recursive calculation in discrete Kalman filter, we can obtain the step-wise posterior distribution of the state variable $\hat{\boldsymbol{c}}[n|n]$ and the fitting error $\hat{\boldsymbol{e}}(n)$ for all steps $n = 1, \cdots, N$.

5.4 Monitoring Tool Wear Progression

To monitor tool wear progression and detect abnormal tool wear progression patterns, we define monitoring statistics from the two Kalman filter estimations. For the state variable, its posterior distribution determines the dynamic linear system in Eq. (5.13), according to a widely acknowledged understanding that a completely specified dynamic linear system without unknown parameters can be drawn from the posterior distribution of the state variables (Petris *et al*., 2009). In discrete Kalman filter, the posterior distribution of the state variable in the current time step is usually set as the prior distribution of the state variable in the next time step. Therefore, monitoring the mean of the posterior distribution of the state variable, $\hat{\boldsymbol{c}}[n|n]$, is an indirect way of monitoring the process dynamics, which is important in revealing tool wear progression patterns. On the other hand, monitoring the fitting error $\hat{\boldsymbol{e}}(n)$, which is the error between the measured tool surface image and the mean of the posterior distribution of the tool pin profile, is equivalent to monitoring the deviation of a tool wear progression from the assumed hierarchical spatio-temporal model. Consequently, it is a natural choice to monitor the two Kalman filter estimations together to reveal tool wear progression patterns.

5.4.1 Monitoring Statistics based on Mahalanobis Distance

In the proposed dynamic linear system, normality is assumed in the initial value of the state variable, $\boldsymbol{c}(1)$, and the error terms, $\boldsymbol{e}(n), n = 1, \cdots, N$. Based on the Kalman filter's recursive process and the property of normal distribution, both $\hat{\boldsymbol{c}}[n|n]$ and $\hat{\boldsymbol{e}}(n)$ follow multivariate normal distribution, $n = 1, \cdots, N$. $\hat{\boldsymbol{c}}[n|n]$ and $\hat{\boldsymbol{e}}(n)$ are assumed to be mutually independent. Since Mahalanobis distance is a measure for the distance between a point to a distribution and can detect outliers in high-dimensional space (Mahalanobis,

1936), we use the step-wise Mahalanobis distances for $\hat{\boldsymbol{c}}[n|n]$ and $\hat{\boldsymbol{e}}(n)$ as the monitoring statistics for tool wear progression. The mean and covariance matrix from the known normal tool wear progressions are needed to calculate the Mahalanobis distance for a testing process. For a set of testing tool wear progressions, the Mahalanobis distance for the mean of the estimated posterior distribution of the state variable at the mth time step is calculated as:

$$d_m^c = (\hat{\boldsymbol{c}}[m|m] - \boldsymbol{\mu}_c)' {\boldsymbol{\Sigma}_c}^{-1} (\hat{\boldsymbol{c}}[m|m] - \boldsymbol{\mu}_c); \tag{5.16}$$

the Mahalanobis distance for the fitting error at the mth time step is calculated as:

$$d_m^e = (\hat{\boldsymbol{e}}(m) - \boldsymbol{\mu}_e)' {\boldsymbol{\Sigma}_e}^{-1} (\hat{\boldsymbol{e}}(m) - \boldsymbol{\mu}_e)\,, \tag{5.17}$$

where $\boldsymbol{\mu}_c$, $\boldsymbol{\mu}_e$, $\boldsymbol{\Sigma}_c$, and $\boldsymbol{\Sigma}_e$ can be estimated from the set of training processes at mth time step.

Due to limited efforts in data collection and tool wear inspection, the number of training processes may be much smaller than the dimension of the two monitored variables. This will cause the ordinary covariance matrices $\boldsymbol{\Sigma}_c$ and $\boldsymbol{\Sigma}_e$ to be not fully ranked, which is known as the ill-posed issue (Tikhonov *et al.*, 2013). As a result, inverses of the ordinary covariance matrices may not exist and the Mahalanobis distances cannot be calculated. Inspired by the regularization method in Chapter 4, we introduce a regularization method in this chapter to solve the issue of ill-posed matrices by adding a regularization term to the ordinary covariance matrix. After regularization, the new covariance matrices will be fully ranked and the regularized Mahalanobis distances will always exist. The regularized covariance matrices can be expressed as:

$$\begin{aligned} \widetilde{\boldsymbol{\Sigma}}_c &= \boldsymbol{\Sigma}_c + \gamma_c \boldsymbol{I} \\ \widetilde{\boldsymbol{\Sigma}}_e &= \boldsymbol{\Sigma}_e + \gamma_e \boldsymbol{I}' \end{aligned} \tag{5.18}$$

where γ_c and γ_e are the regularization parameters for $\hat{\boldsymbol{c}}[m|m]$ and $\hat{\boldsymbol{e}}(m)$, respectively; $\boldsymbol{I}$ is the identity matrix with the same dimension as the ordinary covariance matrix. We use the same γ_c and γ_e values in all time steps. The regularized Mahalanobis distances, i.e., monitoring statistics, can be written as:

$$\tilde{d}_m^c = (\hat{\boldsymbol{c}}[m|m] - \boldsymbol{\mu}_c)' \widetilde{\boldsymbol{\Sigma}}_c^{-1} (\hat{\boldsymbol{c}}[m|m] - \boldsymbol{\mu}_c); \tag{5.19}$$

and

$$\tilde{d}_m^e = (\hat{\boldsymbol{e}}(m) - \boldsymbol{\mu}_e)' \widetilde{\boldsymbol{\Sigma}}_e^{-1} (\hat{\boldsymbol{e}}(m) - \boldsymbol{\mu}_e). \tag{5.20}$$

5.4.2 Thresholds for Tool Wear Progression Monitoring

We assume the tool condition at the initial time step ($n = 1$) is new and fresh, i.e., tool wear progression starts at $n = 2$. We use the 97.5$^{\text{th}}$ percentile and the 2.5$^{\text{th}}$ percentile of the regularized Mahalanobis distances in training processes as the upper and lower thresholds, respectively, for tool wear monitoring. The thresholds are denoted as $\tilde{d}_{0.975}^c, \tilde{d}_{0.025}^c$ and $\tilde{d}_{0.975}^e, \tilde{d}_{0.025}^e$, respectively.

A tool wear progression process in testing will be deemed "abnormal" at the mth time step if at least one of the two mth step monitoring statistics fall outside the thresholds, i.e., if $\tilde{d}_m^c > \tilde{d}_{0.975}^c$, $\tilde{d}_m^c < \tilde{d}_{0.025}^c$, $\tilde{d}_m^e > \tilde{d}_{0.975}^e$, or $\tilde{d}_m^e < \tilde{d}_{0.025}^e$. Since the two monitoring statistics are mutually independent, monitoring the two distances together will ensure that we capture the complete tool wear information to help reveal abnormal tool wear progression patterns

5.4.3 Regularization Parameters

The regularized Mahalanobis distances and hence the performance of the monitoring depend on the value of the regularization parameter γ. We will discuss how γ affects the performance of tool wear progression monitoring and provide suggestion for γ.

Since the mean vector of the posterior distribution of the state variable follows a multivariate normal distribution, the Mahalanobis distance for the mean vectors in the training phase can be regarded as an ellipsoid in high-dimensional space with the center at mean $\boldsymbol{\mu}_c$ and the axis lengths being the eigenvalues of the ordinary covariance matrices $\boldsymbol{\Sigma}_c$. Since the ordinary covariance matrix is ill-posed, some eigenvalues are zero and the corresponding axes are eliminated. Denote the eigenvalues of $\boldsymbol{\Sigma}_c$ as λ_i^c, $i = 1, \cdots, I_1 \times I_2$. After regularization, the eigenvalues of the regularized covariance matrix become:

$$\tilde{\lambda}_i^c = \lambda_i^c + \gamma_c \,. \tag{5.21}$$

Given $\gamma_c > 0$, the new eigenvalues in the regularized covariance matrix $\widetilde{\boldsymbol{\Sigma}}_c$ are equal to or larger than the ordinary eigenvalues. In other words, the Mahalanobis distance ellipsoid is enlarged after regularization. If an observation is deemed "abnormal" using the regularized Mahalanobis distance, then this observation should also be deemed "abnormal" using the ordinary Mahalanobis distance. In this way, the regularization enhances the tool wear monitoring performance. It should be noted that outliers may appear in any direction of the high-dimensional ellipsoid. So, the purpose of regularization is to strengthen the outlier detection with a proper change to the ellipsoid shape, especially to find the outliers along the eliminated directions.

If we choose the value of the regularization parameter to be as large as the largest eigenvalue of the ordinary covariance matrix, i.e., $\gamma_c \approx \max\lambda_i^c, i = 1, \cdots, I_1 \times I_2$, the

Mahalanobis distance ellipsoid will be considerably enlarged in all directions. This value choice emphasizes the outlier detection along the longer axes in the ordinary ellipsoid. Meanwhile, a small regularization value, i.e., $\gamma_c \approx 0, i = 1, \cdots, I_1 \times I_2$, can immediately change the detection ability along the eliminated directions in the ordinary ellipsoid but has almost no improvement on the directions of the longer axes. Therefore, the best γ should strike a balance between the outliers along the longer axes and the outliers in the eliminated directions. A recommended regularization value γ^* is the mean of all eigenvalues in the ordinary covariance matrix:

$$\gamma_c^* = \frac{1}{I_1 I_2} \sum_{i=1}^{I_1 I_2} \lambda_i^c, \tag{5.22}$$

where I_1 and I_2 are the coordinate sizes along the horizontal and vertical dimensions of the tool surface image, respectively, and $I_1 I_2 \times I_1 I_2$ is the size of $\mathbf{\Sigma}_c$.

Similarly, we can derive the regularization parameter for the Mahalanobis distance of the fitting error. The eigenvalues of the regularized covariance matrix $\mathbf{\Sigma}_e$ are:

$$\tilde{\lambda}_i^e = \lambda_i^e + \gamma_e. \tag{5.23}$$

We recommended taking the mean of all eigenvalues in the ordinary covariance matrix $\mathbf{\Sigma}_c$ as the regularization value:

$$\gamma_e^* = \frac{1}{I_1 I_2} \sum_{i=1}^{I_1 I_2} \lambda_i^e ,. \tag{5.24}$$

We will demonstrate the performance of tool wear progression monitoring with different values of the regularization parameter in numerical studies.

5.5 Numerical Study

To demonstrate the proposed method in characterizing and monitoring tool wear progression in micro friction stir welding, we will generate surrogate data to represent different tool wear progression patterns. The model for surrogate data is defined based on the surface measurement of a used μFSW tool (after 400 joints).

5.5.1 μFSW Tool Measurement

The design of μFSW tools is shown in Figure 5.5. It has a stepped cylindrical pin (Wang *et al*., 2018). The pin surface was measured after 400 joints. Figure 5.6 shows the surface profile measurement of the used tool. Both abrasive and adhesive tool wear can be observed from Figure 5.6. The abrasive tool wear is shown as the incomplete circle in the top view. The adhesive wear is shown as the protrusion in the top view. The material of protrusion should flow in from the nearby areas. Both the front view and the 3D view display the material diffusion from center to periphery.

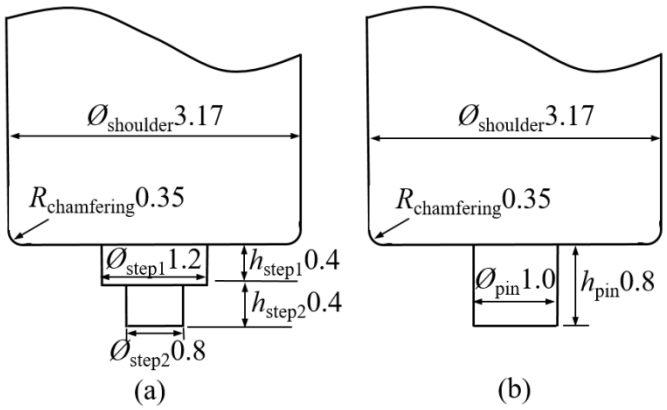


Figure 5.5 Geometries of μFSW tool pin (a) with stepped cylindrical pin and (b) cylindrical pin (unit: mm), adopted from Wang *et al*. (2018)

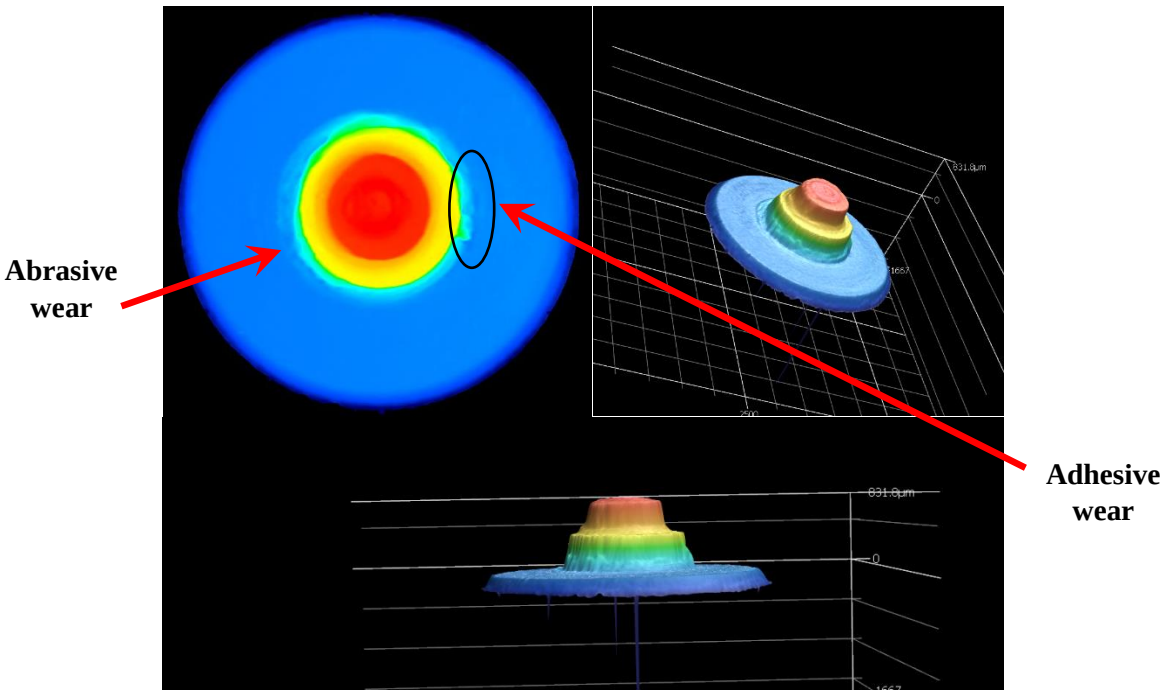


Figure 5.6 Pin surface profile measurement for the used tool in top view, 3D view, and front view

5.5.2 Tool Wear Progression Simulation

For demonstration purpose, we use a simpler tool design Figure 5.5(b) than the measured tool Figure 5.5(a). We generate images of size 60×60 for a μFSW tool with straight cylindrical pin Figure 5.5(b) as the surrogate data of pin surface profile. The cylinder pin is shown as the foreground on the image and the tool shoulder is the background.

5.5.2.1 New Tool

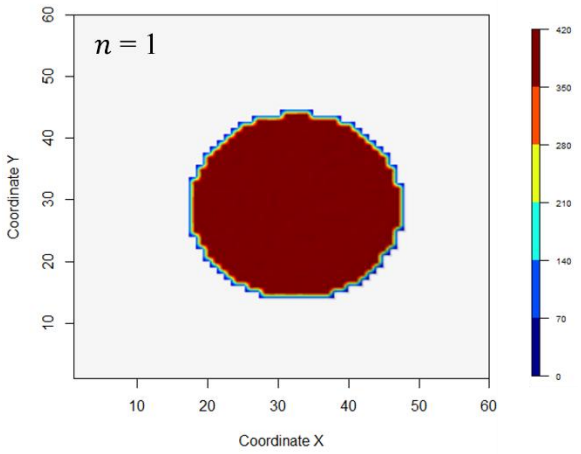
The tool condition at the initial time step ($n = 1$) is new and fresh. We generate the surface profile of a new tool according to the following model. The foreground is round, representing the straight cylinder pin as shown in Figure 5.7. The center of the foreground follows a multivariate normal distribution, $[L_{0x}, L_{0y}]' \sim MVN(\boldsymbol{\mu}_{0L}, \boldsymbol{\Sigma}_{0L})$, where $\boldsymbol{\mu}_{0L} = [\mu_{0x}, \mu_{0y}]' = [30,30]'$ and $\boldsymbol{\Sigma}_{0L} = \text{diag}[\sigma_{0x}^2, \sigma_{0y}^2]' = \text{diag}[1,1]'$. The radius of the foreground circle is constant, $\boldsymbol{R}_0 = [r_{0x}, r_{0y}]' = [15,15]'$. The surface height of

the foreground is $\boldsymbol{y_s}(1)\sim N(400, 21)$ and the height of the background is $\boldsymbol{y_s}(1)\sim N(0, 21)$, showing that the pin surface is about 400 units higher than the shoulder. The variance 21 is the average variance in the measured tool image displayed in Figure 5.6. Figure 5.7(a) shows a simulated new tool surface profile.

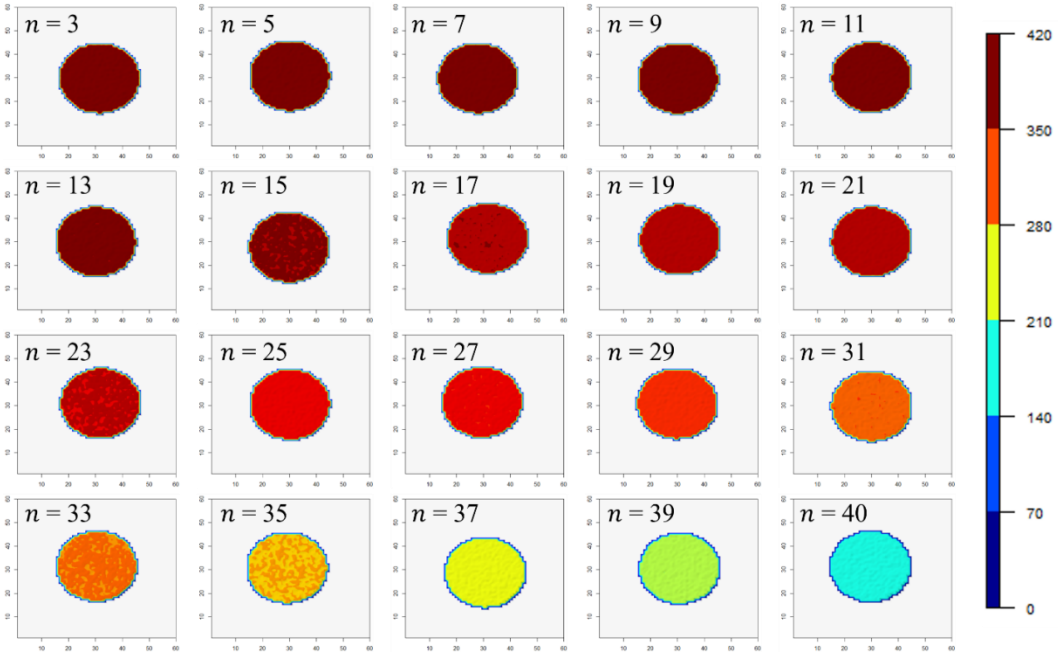
5.5.2.2 Normal Tool Wear Progression

Tool wear progression starts at $n = 2$. We define a normal tool wear progression pattern and use these data as the training data to the proposed method. In the normal tool wear progression, the tool pin profile height reduces proportionally to the designed useful life, while the pin shape remains unchanged. The designed useful life of the tool is $N = 40$.

The first order autoregression – AR(1) model is adopted to describe the dynamics of tool wear progression. The tool pin profile height change at the nth step is $\delta_0(n)$, $\delta_0(n) = 10 + 1.1\delta_0(n-1) + \tau(n)$, $n = 2, \cdots, 40$, where $\tau(n)\sim N(0, 0.5)$. The coefficient 1.1 implies that the AR(1) is a nonstationary process; the constant 10 is selected to keep all the generated $\delta_0(n)$'s positive. $\delta_0(n)$ is then scaled to $\delta_0'(n)$ in range $[0, 200]$, i.e., $\delta_0'(1) = 0$ and $\delta_0'(40) = 200$. The height of tool pin profile in a normal tool wear progression is: $\boldsymbol{y}_s^0(n)\sim N\left(400 - \delta_0'(n), \left({\delta_0'(n)}/{100}\right)^2\right)$. The background stays the same as the new tool, i.e., $\boldsymbol{y}_s^0(n)\sim N(0, 21)$ if the coordinate grid $\boldsymbol{s}$ is outside the foreground circle. Figure 5.7(b) shows the simulated tool surface images of a normal tool wear progression. Average pin profile heights gradually reduce from 400 at time stamp 1 to 140 at the termination step 40 but the size and shape of the pin part always keep the same.



(a) New tool



(b) Normal tool wear progression at the *n*th time step

Figure 5.7 Example of a simulated (a) new tool and (b) normal tool wear progression at the *n*th time step, pin profile height shown in color map

5.5.2.3 Three Abnormal Tool Wear Progressions

Scenario 1: Faster height reduction

We assume that the tool pin profile height reduces faster than the height reduction in the normal tool wear process and, as a result, the pin will reach end-of-life height in less than 40 steps. We simulate 7 cases with different tool life N_1 (termination steps), as listed in Table 5.1. Specifically, all pins have the same shape at the termination step N_1; a smaller N_1 represents a faster tool wear progression. The tool pin profile height change at the nth step is $\delta_1(n), n = 1, \cdots, N_1$. The height of tool pin profile in this scenario is $\boldsymbol{y_s^1}(n) \sim N\left(400 - \delta_1'(n), \left({\delta_1'(n)}/{100}\right)^2\right)$. Figure 5.8 shows the simulated tool surface images in Scenario 1. The background stays the same as the new tool, i.e., $\boldsymbol{y_s^1}(n) \sim N(0, 21)$ if the coordinate grid $\boldsymbol{s}$ is outside the foreground circle. The AR(1) model is also the same as in the normal tool wear process.

Table 5.1 Tool life in scenario 1

	Normal	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7
Designed tool life in steps, N_1	40	38	36	34	32	30	28	26

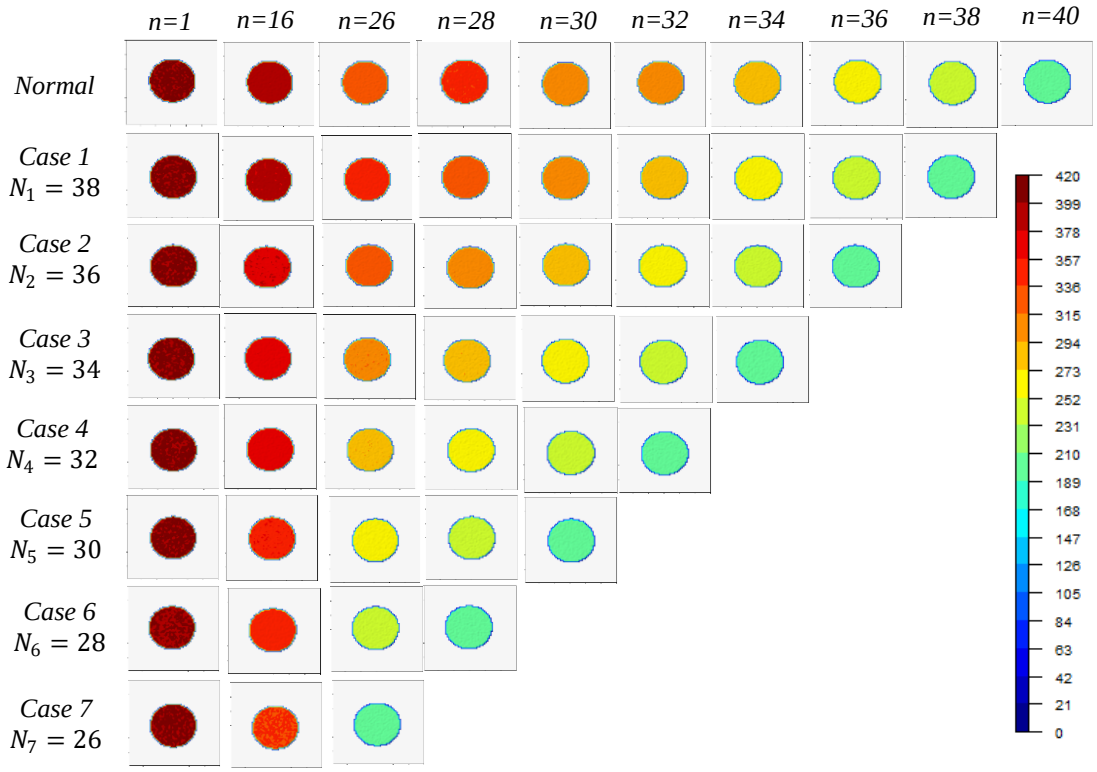


Figure 5.8 Abnormal tool wear progression in scenario 1

Scenario 2: Abrasive wear

Abrasive wear is caused by the displacement of material from a solid surface due to hard particles or protuberances sliding along the surface (Weinberger *et al.*, 2008). For abrasive wear, we assume that the round tool pin is changed to a smaller round shape. The radius change at the nth step is $\delta_2(n) = 10 + 1.1\delta_2(n-1) + \tau(n), n = 1, \cdots, 40$, where $\tau(n) \sim N(0, 0.5)$. $\delta_2(n)$ is then scaled to $\delta_2'(n)$ in range $[0, \delta_2'(40)]$ with 5 cases,

i.e., $\delta_2'(1) = 0$ and $\delta_2'(40) = \{2,4,6,8,10\}$. The radius of the foreground circle (pin) at the nth step is $[r_{2x}(n), r_{2y}(n)]^T = [r_{0x} - \delta_2'(n), r_{0y} - \delta_2'(n)]^T$. The final pin radius at step 40 for the 5 cases are displayed in Table 5.2. The tool pin profile height reduces over time with the same rate as the normal tool wear progression. Figure 5.9 shows the simulated tool surface images in Scenario 2. A smaller final pin radius represents more severe abrasive wear to the tool and therefore a shorter useful life. The degree of the abrasive wear increases from Case 1 to Case 5.

Table 5.2 The pin radius at $N = 40$ in scenario 2

Normal: $r_{0x} = r_{0y} = 15$	**Case 1**	**Case 2**	**Case 3**	**Case 4**	**Case 5**
Pin radius, $r_{2x} = r_{2y} = r$	13	11	9	7	5

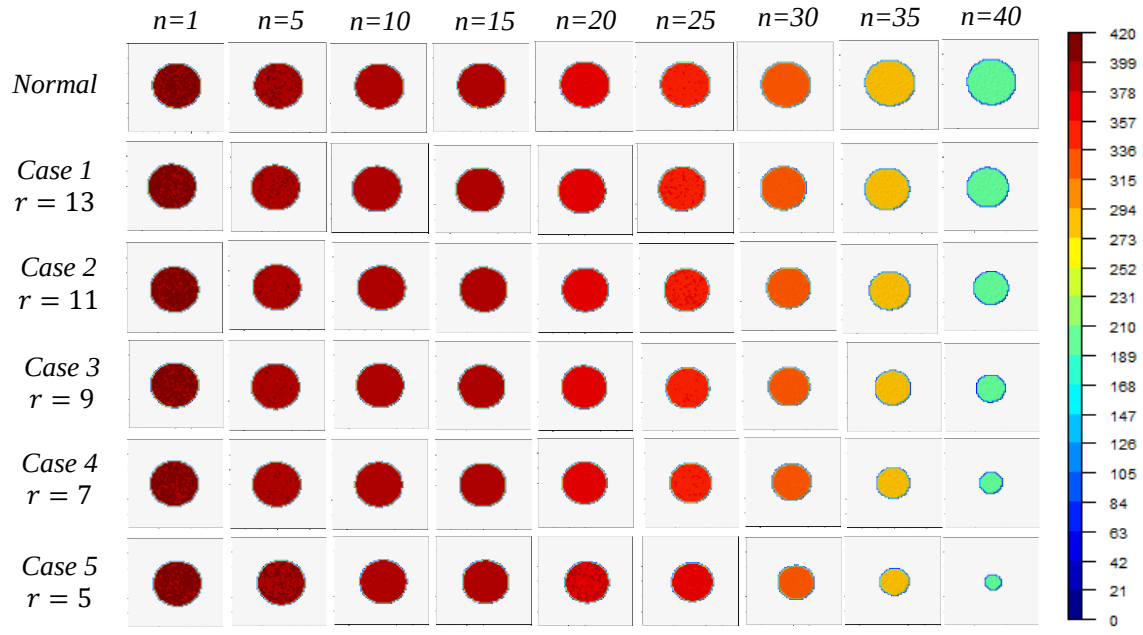


Figure 5.9 Abnormal tool wear progression in scenario 2

Scenario 3: Adhesive wear

Adhesive wear is caused by moving of material from one surface to another either permanently or temporarily during relative motion in μFSW welding (Weinberger *et al*., 2008). For adhesive wear, we assume the round pin gradually changes to an ellipse shape with certain eccentricity. We define $\delta_3(n) = 10 + 1.1\delta_3(n-1) + \tau(n)$, $n = 1, \cdots, 40$, where $\tau(n) \sim N(0, 0.5)$. $\delta_3(n)$ is scaled to $\delta_3'(n)$ in range $[0, \delta_3'(40)]$ with 7 cases, i.e., $\delta_3'(1) = 0$ and $\delta_3'(40) = \{0.07, 0.25, 0.5,\ 0.875, 1.5, 2.75, 6.5\}$. The radius of the pin at the nth step is $\left[r_{3x}(n), r_{3y}(n)\right]^T = \left[\left(1 + \delta_3'(n)\right) r_{0x}, \frac{r_{0y}}{1+\delta_3'(n)}\right]^T$. The final short/long axis lengths for the ellipse at the termination step for the 7 cases are displayed in Table 5.3. The tool pin profile height reduces over time with the same rate as the normal tool wear progression. Figure 5.10 shows the simulated tool surface images in Scenario 3. A larger eccentricity value represents more severe adhesive wear to the tool and therefore a shorter useful life. The tool useful life decreases from Case 1 to Case 7.

Table 5.3 The short and long axis lengths and eccentricity at $N = 40$ in scenario 3

	Normal	**Case 1**	**Case 2**	**Case 3**	**Case 4**	**Case 5**	**Case 6**	**Case 7**
Short axis length	15	14	12	10	8	6	4	2
Long axis length	15	16.1	18.75	22.5	28.125	37.5	60	112.5
Eccentricity	0	0.491	0.768	0.896	0.959	0.987	0.998	0.99984

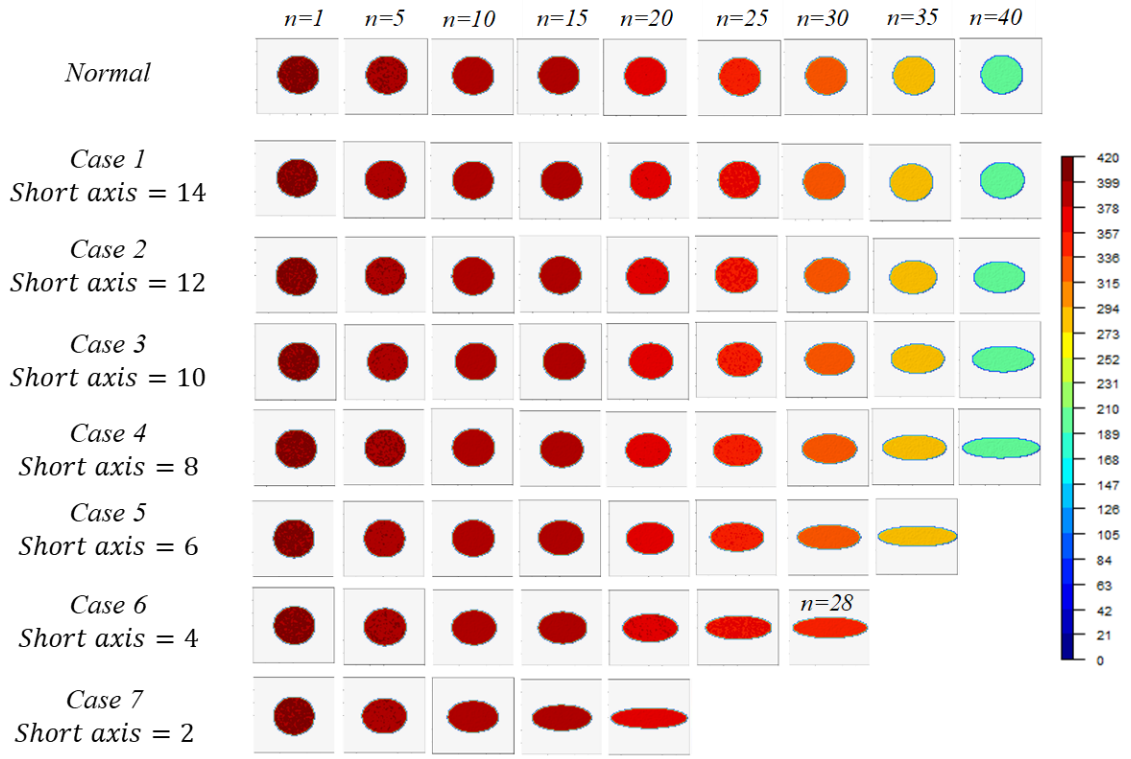


Figure 5.10 Abnormal tool wear progression in scenario 3

5.5.3 Model Fitting and Tool Wear Monitoring Results

We generated 120 tool wear processes for the normal tool wear progression scenario and 16 processes each in Scenarios 1, 2, and 3. We simulate tool surface images for 40 steps and monitor the first 25 steps. The rationale is that the tool wear is quite significant after 25 steps and it can be easily detected by various simple methods. We first obtain the posterior distributions of the state variable for the training tool wear processes $\hat{\boldsymbol{c}}_0^k[n|n]$ and the fitting error $\hat{\boldsymbol{e}}_0^k(n)$, $k = 1, \cdots, 120$. We then calculate the regularized Mahalanobis distance for training data and obtain the decision thresholds $\tilde{d}^c_{0.975}, \tilde{d}^c_{0.025}$ and $\tilde{d}^e_{0.975}, \tilde{d}^e_{0.025}$. Next, we test the three abnormal tool wear scenarios by fitting the tool surface profile with the proposed hierarchical spatio-temporal model and obtaining the two Kalman filter estimations. They are represented as $\hat{\boldsymbol{c}}_1^l[n|n]$, $\hat{\boldsymbol{c}}_2^l[n|n]$, $\hat{\boldsymbol{c}}_3^l[n|n]$ and

$\hat{\boldsymbol{e}}_1^l(n)$, $\hat{\boldsymbol{e}}_2^l(n)$, $\hat{\boldsymbol{e}}_3^l(n)$, where $l = 1, \cdots, 16$ denotes the number of processes in each abnormal tool wear scenario.

As stated in Section 5.4.3, we recommend using the mean of eigenvalues of the ordinary covariance matrix from all training tool wear processes as the regularization parameter, $\{\gamma_c, \gamma_e\} = \left\{\frac{1}{3600}\sum_{i=1}^{3600}\lambda_i^c, \frac{1}{3600}\sum_{i=1}^{3600}\lambda_i^e\right\}$. The monitoring results are shown in Table 5.4. Three abnormal tool wear progression scenarios are listed in Table 5.4 (a), (b), and (c), respectively. In each table, the tool wear severity increases from Case 1 to Case 7. We quantify the monitoring performance in three measures: (1) The first time step that the abnormal tool wear progression is detected, which indicates the timeliness of the detection; (2) The fraction of time steps (out of 24, from the 2nd to the 25th time step) in which the abnormal tool wear progression is detected, which indicates the confidence in detection; (3) The Type II error rate, which is the rate that an abnormal tool wear progression is not detected, indicating the power of detection.

Table 5.4 (a) Tool wear progression monitoring results for Scenario 1 with the recommended regularization parameter

Monitoring Statistics	**Case 1**	**Case 2**	**Case 3**	**Case 4**	**Case 5**	**Case 6**	**Case 7**
	more severe tool wear →						
First time step that the abnormal tool wear progression is detected							
state variable	2	2	2	2	2	2	2
fitting error	2	2	2	2	2	2	2
state variable + fitting error	**2**	**2**	**2**	**2**	**2**	**2**	**2**
Fraction of time steps in which the abnormal tool wear progression is detected							
the state variable	100.0%	96.4%	97.4%	96.4%	98.2%	97.7%	100.0%
The error variable	100.0%	76.3%	77.6%	81.0%	81.3%	79.7%	91.7%
state variable + fitting error	**100.0%**	**99.5%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**
Rate to detect the abnormal tool wear progression as the normal progression							
the state variable	0	0	0	0	0	0	0
The error variable	0	0	0	0	0	0	0
state variable + fitting error	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Table 5.4 (b) Tool wear progression monitoring results for Scenario 2 with the recommended regularization parameter

Monitoring Statistics	**Case 1**	**Case2**	**Case3**	**Case4**	**Case5**
	more severe tool wear →				
First time step that the abnormal tool wear progression is detected					
state variable	2	2	2	2	2
fitting error	2	2	2	2	2
state variable + fitting error	**2**	**2**	**2**	**2**	**2**
Fraction of time steps in which the abnormal tool wear progression is detected					
the state variable	100.0%	100.0%	100.0%	100.0%	100.0%
The error variable	100.0%	100.0%	100.0%	100.0%	100.0%
state variable + fitting error	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**
Rate to detect the abnormal tool wear progression as the normal progression					
the state variable	0	0	0	0	0
The error variable	0	0	0	0	0
state variable + fitting error	**0**	**0**	**0**	**0**	**0**

Table 5.4 (c) Tool wear progression monitoring results for Scenario 3 with the recommended regularization parameter

Monitoring Statistics	**Case 1**	**Case 2**	**Case 3**	**Case 4**	**Case 5**	**Case 6**	**Case 7**
	more severe tool wear →						
First time step that the abnormal tool wear progression is detected							
state variable	2	2	2	2	2	2	2
fitting error	2	2	2	2	2	2	2
state variable + fitting error	**2**	**2**	**2**	**2**	**2**	**2**	**2**
Fraction of time steps in which the abnormal tool wear progression is detected							
the state variable	99.5%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
The error variable	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
state variable + fitting error	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**
Rate to detect the abnormal tool wear progression as the normal progression							
the state variable	0	0	0	0	0	0	0
The error variable	0	0	0	0	0	0	0
state variable + fitting error	**0**	**0**	**0**	**0**	**0**	**0**	**0**

For comparison purpose, we show the monitoring results when we use only the state variable or the error variable, or the proposed integration of state variable and error variable. It can be noticed from Table 5.4 that the results from the integrated decision making is generally better than using a single statistic, demonstrating that the two Kalman filter estimations are complementary to each other. Using the integrated statistics, the progression abnormality for all three scenarios can be detected from the 2nd time step, which is the first step after abnormality occurs, showing timeliness of the detection. Under Scenarios 2 and 3, the progression abnormality can be detected 100% in all 24 steps (from the 2nd step to the 25th step); the progression abnormality can be found in more than 70% of the steps under Scenario 1. This shows a confidence in the detection. Moreover, the Type II error rate is 0 in all three scenarios, showing a high detection

power. We can conclude from Table 5.4 that, with the recommended regularization parameter, the proposed hierarchical spatio-temporal model can effectively capture tool wear progression patterns and the proposed monitoring strategy can detect abnormal tool wear progressions effectively and efficiently.

To further demonstrate the choice of the regularization parameter, we perform the same procedure with two alternatives of $\{\gamma_c, \gamma_e\}$. (1) A large regularization parameter: we adopt the largest eigenvalue of the ordinary covariance matrix from all training tool wear processes, $\{\gamma_c, \gamma_e\} = \{\max \lambda_i^c,\ \max \lambda_i^e\},\ i = 1, \cdots, 3600$. (2) A small regularization parameter: we adopt the smallest positive eigenvalue of the ordinary covariance matrix from all training tool wear processes, $\{\gamma_c, \gamma_e\} = \{\min \lambda_i^c,\ \min \lambda_i^e\}$ for $\forall\ \lambda_i^c > 0\ and\ \lambda_i^e > 0, i = 1, \cdots, 3600$. The monitoring results are shown in Table 5.5 and Table 5.6, respectively.

Tables 5.4~5.6 show that the proposed method with integrated monitoring statistics (state variable + fitting error) can detect the majority of the abnormal tool wear progression patterns in all cases. Using the integrated monitoring statistics, the fraction of time steps (out of 24, from the 2nd time step to the 25th) in which the abnormal tool progression can be detected is almost always better than the results by using one estimation from Kalman filter; at least 59.6% (in Table 5.5(a) Case 1) of the time steps can be detected, which shows good performance with consistency; the averaged first step that the abnormal tool progression can be detected is almost 2, which means as soon as an abnormal tool wear progression pattern occurs, the proposed method can detect it immediately; the Type II error rate is 0, showing no mis-detection.

The comparison of Tables 5.4~5.6 shows the different performance when using different regularization parameters. Adopting a large regularization parameter, the performance by monitoring the state variable is very good in all three scenarios, yet the performance does not monotonically increase as the abnormal wear gets more severe, which is counterintuitive. The performance by monitoring the fitting error is poor. Take Scenario 1's Case 5 for example, none of the abnormal tool wear processes from the 2^{nd} to the 25^{th} time steps could be detected; the Type II error rates in monitoring the error variable in all three scenarios are also very high, suggesting a large test error. Since the fitting error is a very important measure of the performance of the Kalman filter, a poor performance from the error monitoring undermines the validity of the estimation from the discrete Kalman filter. Therefore, we do not recommend taking the largest eigenvalue of the ordinary covariance matrix as the regularization value for regularized Mahalanobs distance monitoring.

Table 5.4 and Table 5.6 display the monitoring performance by setting the regularization parameter as the mean of all eigenvalues and as the smallest positive eigenvalue of the ordinary covariance matrix, respectively. Since most of the eigenvalues in the ordinary covariance matrix are zero, the mean of all eigenvalues is smaller than the smallest positive eigenvalue. In the tool wear progression monitoring for Scenarios 2 and 3, both Tables 5.4 and 5.6 provide good monitoring performance and Table 5.4 has slightly better results than those in Table 5.6. But results in Table 5.4 significantly outperforms those in Table 5.6 for Scenario 1. Therefore, we can conclude the recommended regularization parameter provides the best monitoring performance.

Table 5.5 (a) Tool wear progression monitoring results for Scenario 1 with a large regularization parameter

Monitoring Statistics	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7
	more severe tool wear →						
First time step that the abnormal tool wear progression is detected							
state variable	2	2	2	2	2	2	2
fitting error	14	19	14	11	–	14	13
state variable + fitting error	**2**	**2**	**2**	**2**	**2**	**2**	**2**
Fraction of time steps in which the abnormal tool wear progression is detected							
state variable	57.6%	62.5%	74.2%	76.0%	78.9%	83.3%	89.6%
fitting error	4.2%	6.8%	7.0%	6.5%	0.0%	12.8%	12.5%
state variable + fitting error	**59.6%**	**65.6%**	**75.8%**	**79.2%**	**78.9%**	**84.9%**	**90.9%**
Rate to detect the abnormal tool wear progression as the normal progression							
state variable	0	0	0	0	0	0	0
fitting error	0.5	0.6875	0.5	0.375	1	0.4375	0.375
state variable + fitting error	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Table 5.5 (b) Tool wear progression monitoring results for Scenario 2 with a large regularization parameter

Monitoring Statistics	Case 1	Case2	Case3	Case4	Case5
	more severe tool wear →				
First time step that the abnormal tool wear progression is detected					
state variable	2	2	2	2	2
fitting error	11	12	6	2	4
state variable + fitting error	**2**	**2**	**2**	**2**	**2**
Fraction of time steps in which the abnormal tool wear progression is detected					
state variable	99.7%	99.5%	100.0%	100.0%	100.0%
fitting error	12.8%	9.6%	20.1%	61.2%	21.1%
state variable + fitting error	**99.7%**	**99.5%**	**100.0%**	**100.0%**	**100.0%**
Rate to detect the abnormal tool wear progression as the normal progression					
state variable	0	0	0	0	0
fitting error	0.3125	0.375	0.125	0	0.0625
state variable + fitting error	**0**	**0**	**0**	**0**	**0**

Table 5.5 (c) Tool wear progression monitoring results for Scenario 3 with a large regularization parameter

Monitoring Statistics	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7
	more severe tool wear →						
First time step that the abnormal tool wear progression is detected							
state variable	2	2	2	2	2	2	2
fitting error	13	4	6	13	9	2	3
state variable + fitting error	2	2	2	2	2	2	2
Fraction of time steps in which the abnormal tool wear progression is detected							
state variable	90.9%	96.6%	100.0%	100.0%	100.0%	100.0%	100.0%
fitting error	11.7%	17.2%	13.5%	17.7%	10.4%	75.0%	36.7%
state variable + fitting error	**91.1%**	**97.1%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**
Rate to detect the abnormal tool wear progression as the normal progression							
state variable	0	0	0	0	0	0	0
fitting error	0.4375	0.0625	0.125	0.4375	0.25	0	0
state variable + fitting error	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Table 5.6 (a) Tool wear progression monitoring results for Scenario 1 with a small regularization parameter

Monitoring Statistics	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7
	more severe tool wear →						
First time step that the abnormal tool wear progression is detected							
state variable	2	2	2	2	2	2	2
fitting error	2	2	2	2	2	2	2
state variable + fitting error	**2**	**2**	**2**	**2**	**2**	**2**	**2**
Fraction of time steps in which the abnormal tool wear progression is detected							
state variable	57.6%	62.5%	74.2%	76.0%	78.9%	83.3%	89.6%
fitting error	51.8%	70.8%	71.4%	74.2%	78.9%	80.5%	82.6%
state variable + fitting error	**70.3%**	**79.2%**	**82.0%**	**84.6%**	**87.0%**	**89.8%**	**94.3%**
Rate to detect the abnormal tool wear progression as the normal progression							
state variable	0	0	0	0	0	0	0
fitting error	0	0	0	0	0	0	0
state variable + fitting error	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Table 5.6 (b) Tool wear progression monitoring results for Scenario 2 with a small regularization parameter

Monitoring Statistics	Case 1	Case2	Case3	Case4	Case5
	more severe tool wear →				
First time step that the abnormal tool wear progression is detected					
state variable	2	2	2	2	2
fitting error	2	2	2	2	2
state variable + fitting error	**2**	**2**	**2**	**2**	**2**
Fraction of time steps in which the abnormal tool wear progression is detected					
state variable	99.7%	99.5%	100.0%	100.0%	100.0%
fitting error	100.0%	100.0%	100.0%	100.0%	100.0%
state variable + fitting error	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**
Rate to detect the abnormal tool wear progression as the normal progression					
state variable	0	0	0	0	0
fitting error	0	0	0	0	0
state variable + fitting error	**0**	**0**	**0**	**0**	**0**

Table 5.6 (c) Tool wear progression monitoring results for Scenario 3 with a small regularization parameter

Monitoring Statistics	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7
	more severe tool wear →						
First time step that the abnormal tool wear progression is detected							
state variable	2	2	2	2	2	2	2
fitting error	2	2	2	2	2	2	2
state variable + fitting error	**2**	**2**	**2**	**2**	**2**	**2**	**2**
Fraction of time steps in which the abnormal tool wear progression is detected							
state variable	90.9%	96.6%	100.0%	100.0%	100.0%	100.0%	100.0%
fitting error	97.9%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
state variable + fitting error	**97.9%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**
Rate to detect the abnormal tool wear progression as the normal progression							
state variable	0	0	0	0	0	0	0
fitting error	0	0	0	0	0	0	0
state variable + fitting error	**0**	**0**	**0**	**0**	**0**	**0**	**0**

To sum up, the proposed method with the recommended regularization parameter provides the best results in tool wear progression monitoring. In this numerical study, the covariance matrix is badly ill-posed: the variable dimension is 3600 while the number of training processes is only 120, $1/30$ of the variable dimension. As a result, many of the eigenvalues in the ordinary covariance matrix are zero. Comparing to the smallest positive eigenvalue, setting the regularization parameter as the mean of all eigenvalues can make a better balance among all eigenvalues and therefore, provide a better outlier detection improvement along all directions of the ordinary Mahalanobis distance ellipsoid. Therefore, the average of all eigenvalues is recommended for the regularized Mahalanobis distance for tool wear monitoring purpose.

5.6 Conclusion

In μFSW, tool wear usually occurs dynamically and spatially. Despite the significance of tool wear on process dynamics and part quality, there is a lack of effective μFSW tool condition monitoring system, inspiring us to develop a new method to effectively characterize tool wear and monitor tool wear progression for μFSW. Practically, tool wear can be measured indirectly as a high-resolution image of the surface, but the tool wear mechanism depends on the everchanging usage condition during welding process. In Chapter 5, we develop a novel hybrid hierarchical spatio-temporal model to characterize tool wear by integrating the image data and the physics of the process. Based on the developed spatio-temporal model, we can monitor tool wear progression and detect abnormal progression patterns in time and with confidence.

The tool surface measurement images taken at different stages of the tool has both spatial and temporal features. To effectively characterize such dependencies, we first

build a Bayesian inference based hierarchical spatio-temporal model, in which the first level is a linear model for the high-resolution tool pin profile images and the second level is a physics-based advection-diffusion model for the latent (unobserved) welding temperature distribution. With the hybrid hierarchical structure, we characterize the spatial evolution of the tool wear over time.

We then discretize the process variable from the continuous time series into the discretized time steps with the finite element approximations. The advection-diffusion equation is converted to a dynamic state space equation and the hierarchical model is then transferred to a dynamic linear system. Discrete Kalman filter is adopted to fit the dynamic linear system and estimate the posterior distributions of the discretized temperature distribution and the tool pin profile. The Kalman filter provides two important estimations. One is the mean of the posterior distribution of the discretized temperature distribution and the other is the fitted error between the mean of the posterior distribution of the tool pin profile and the measured tool pin profile.

We propose a regularized Mahalanobis distance from the Kalman filter estimations to develop the tool wear progression monitoring statistics. Decision thresholds for the two estimations are first calculated from the normal tool wear progressions. The testing processes, i.e., tool wear progressions, in three wear mechanisms are monitored with the proposed integrated decision making. Regularization method is adopted to handle the ill-posed issue in the ordinary covariance matrix of the training processes. To evaluate the performance of tool wear monitoring, we define three measures to quantify the timeliness of detection, the confidence of decision, and the detection power. A recommended regularization parameter is provided to optimize the

monitoring performance. Numerical studies based on three tool wear mechanisms are implemented to demonstrate the feasibility and effectiveness of our proposed method.

The proposed method in this chapter demonstrates its uniqueness in combining the high-dimensional tool wear images with the physics-based μFSW process, its feasibility in modeling the dynamic process evolution in spatial dimension over time, as well as its effectiveness and timeliness in detecting abnormal tool wear progression patterns. The proposed method can also be extended to similar welding processes with tool inspection capability and similar process conditions. The method developed in this chapter can potentially contribute to innovations in joining lightweight materials. Although micro friction stir welding is the motivation behind this chapter, the method development can be further extended to other manufacturing processes.

CHAPTER 6 CONCLUSION AND FUTURE RESEARCH

6.1 Conclusion

In this dissertation, we develop process analytics methods for process monitoring and tool condition monitoring for the joining of lightweight materials.

Motivated by joining lightweight materials via Friction Stir Blind Riveting (FSBR), this research enables *in-situ* NDE of FSBR by establishing the relationship between *in-situ* process information and joint quality. The proposed methods effectively integrate tensor decomposition, feature selection, and decision-making. The performance of the proposed methods is evaluated by the correct classification rate. The proposed methods are compared to competitors in numerical studies and FSBR case study.

Motivated by joining lightweight materials via Micro Friction Stir Welding (μFSW), this research characterizes the dynamic tool wear propagation in μFSW by fitting a hierarchical spatio-temporal model to the sequence of tool surface measurement images, which enables early detection of abnormal tool wear progression patterns in μFSW. The proposed method is demonstrated with various surrogate data in numerical studies.

In Chapter 3, we investigate the effectiveness of PCA-based unsupervised tensor decomposition methods in extracting features from nonlinear multi-sensor profile data. The extract features are combined with engineering knowledge to form a set of enriched features for the FSBR process. Then, sparse group lasso regression is adopted to select the most significant features from this enriched process information. These selected features are fed into five classifiers, whose results are fused to provide the final quality

evaluation. The optimal weights are obtained by maximizing the correct classification rate in training dataset.

In Chapter 4, we focus on NDE with small training sample size but known class information, which is common in manufacturing applications. With class information known, LDA-based supervised tensor decomposition methods with regularization (R-UMLDA) is utilized to handle the small sample size scenario. Since the features extracted from R-UMLDA are sensitive to the regularization parameter, ensemble learning is integrated with R-UMLDA to stabilize the feature extraction and provide high accuracy in quality evaluation. We develop optimal integration by selecting features to increase the base learner diversity so that ensemble learning performance can be enhanced. A heuristic adaptive method is also developed to obtain the optimal regularization parameter for R-UMLDA.

In Chapter 5, we develop a hybrid hierarchical spatio-temporal model for the time-ordered tool surface measurement images to characterize the dynamic tool wear propagation in μFSW. The model is developed in a hierarchical Bayesian structure with the first level being a data-driven regression model for the high-resolution tool pin profile images and the second level being a physics-based advection-diffusion model for the welding temperature distribution. Kalman filter is adopted to estimate the posterior distributions of the state variable (temperature distribution) and the error between the measured tool surface image and the predicted images. To monitor tool wear progression, monitoring statistics are defined from the two estimations using Mahalanobis distance. Regularization is added to the monitoring statistics to handle the ill-posed issue in Mahalanobis distance calculation. A recommended value for the regularization parameter

is provided to optimize the monitoring performance. Thresholds are defined to detect abnormal tool wear progression patterns. Three abnormal tool wear progression patterns are simulated to demonstrate the effectiveness of the proposed method.

6.2 Future Research

Despite the satisfactory performance of the proposed methodologies in this dissertation, there is still some room for further improvements.

For the research on process monitoring and tool wear monitoring, it would be beneficial if the monitoring result can guide process adjustment or corrective actions. This requires that we quickly identify the root cause behind a bad-quality part, an abnormal process condition, or an abnormal tool wear progression pattern, and then quickly identify the optimal control action to either fix the part or adapt the process to such unexpected abnormal conditions. Future study should investigate how to use process/tool wear monitoring to support diagnosis and optimization capabilities.

The integration of *in-situ* NDE and tool condition monitoring is also an important direction for future research. A process-tool-part quality relationship can help ensure process robustness, tool condition, and part quality. This relationship can also be used for smart tool planning to extend tool usage life, increase overall productivity, and reduce cost.

By using regularization, our research development is able to handle small sample size problem. However, there are situations when we don’t even have a small sample. This could arise when we have a new process, new materials, new product design, new quality requirement, etc., leading to new root causes and new tool wear patterns. It will

be interesting to empower process/tool wear monitoring with learning capabilities so that the efforts spent in model training is minimized.

With the widespread deployment of sensors and Internet of Things, more manufacturing data in more complicated formats are becoming available. Deep learning provides advanced analytics tools for processing and analyzing big manufacturing data. The data analytics methods developed in this dissertation fall into the traditional machine learning category, while deep learning will be more adequate in handling voluminous data. Future research can investigate deep learning based methods to improve system performance in manufacturing.

References

ABU-ZAHRA, N. H. & NAYFEH, T. H. 1997. Calibrated method for ultrasonic on-line monitoring of gradual wear during turning operations. International Journal of Machine Tools and Manufacture, 37, 1475-1484.

ADOLFSSON, S., BAHRAMI, A., BOLMSJÖ, G. & CLAESSON, I. 1999. On-line quality monitoring in short-circuit gas metal arc welding. WELDING JOURNAL-NEW YORK-, 78, 59-s.

AHMAD, M., NUAWI, M. Z., ABDULLAH, S., WAHID, Z., KARIM, Z. & DIRHAMSYAH, M. 2015. Development of tool wear machining monitoring using novel statistical analysis method, I-kaz™. Procedia Engineering, 101, 355-362.

ALTMEYER, J., DOS SANTOS, J. & AMANCIO-FILHO, S. 2014. Effect of the friction riveting process parameters on the joint formation and performance of Ti alloy/short-fibre reinforced polyether ether ketone joints. Materials & Design, 60, 164-176.

BASIR, O. & YUAN, X. 2007. Engine fault diagnosis based on multi-sensor information fusion using Dempster–Shafer evidence theory. Information Fusion, 8, 379-386.

BERLINER, L. M. 2003. Physical-statistical modeling in geophysics. Journal of Geophysical Research: Atmospheres, 108.

BERLINER, L. M., WIKLE, C. K. & CRESSIE, N. 2000. Long-lead prediction of Pacific SSTs via Bayesian dynamic modeling. Journal of climate, 13, 3953-3968.

BERMINGHAM, M. L., PONG-WONG, R., SPILIOPOULOU, A., HAYWARD, C., RUDAN, I., CAMPBELL, H., WRIGHT, A. F., WILSON, J. F., AGAKOV, F. & NAVARRO, P. 2015. Application of high-dimensional feature selection: evaluation for genomic prediction in man. Scientific reports, 5, 10312.

BHOWMICK, A. 2010. Prediction of friction stir weld quality using self-organized maps. Dissertation, Aukland University of Technology, Aukland, New Zealand.

BISWAS, P., A KUMAR, D. & MANDAL, N. 2012. Friction stir welding of aluminum alloy with varying tool geometry and process parameters. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 226, 641-648.

BLAGA, L., DOS SANTOS, J., BANCILA, R. & AMANCIO-FILHO, S. 2015. Friction Riveting (FricRiveting) as a new joining technique in GFRP lightweight bridge construction. Construction and Building Materials, 80, 167-179.

BOLDSAIKHAN, E. 2008. The use of feedback forces for nondestructive evaluation of friction stir welding. South Dakota School of Mines and Technology, Rapid City.

BOLDSAIKHAN, E., BHARAT, J., LOGAR, A., CORWIN, E., JANES, M. & ARBEGAST, W. A phase space approach to detecting volumetric defects in friction stir welding. Proceedings of 8th International Friction Stir Welding Symposium, Timmendorfer Strand, Germany, 2010.

BOLDSAIKHAN, E., CORWIN, E., LOGAR, A. & ARBEGAST, W. Neural network evaluation of weld quality using FSW feedback data. Proceedings of Friction Stir Welding, 6th International Symposium, 2006.

BOLDSAIKHAN, E., CORWIN, E. M., LOGAR, A. M. & ARBEGAST, W. J. 2011. The use of neural network and discrete Fourier transform for real-time evaluation of friction stir welding. Applied soft computing, 11, 4839-4846.

BOOTHROYD, G. 1988. Fundamentals of metal machining and machine tools, Crc Press.

BREIMAN, L. 1996. Bagging predictors. Machine learning, 24, 123-140.

BREIMAN, L. 2001. Random forests. Machine learning, 45, 5-32.

BREIMAN, L. 2017. Classification and regression trees, Routledge.

CALDER, C. A. 2008. A dynamic process convolution approach to modeling ambient particulate matter concentrations. Environmetrics: The official journal of the International Environmetrics Society, 19, 39-48.

CAMELETTI, M., IGNACCOLO, R. & BANDE, S. 2011. Comparing spatio-temporal models for particulate matter in Piemonte. Environmetrics, 22, 985-996.

CAMELETTI, M., LINDGREN, F., SIMPSON, D. & RUE, H. 2013. Spatio-temporal modeling of particulate matter concentration through the SPDE approach. AStA Advances in Statistical Analysis, 97, 109-131.

CASTEJÓN, M., ALEGRE, E., BARREIRO, J. & HERNÁNDEZ, L. 2007. On-line tool wear monitoring using geometric descriptors from digital images. International Journal of Machine Tools and Manufacture, 47, 1847-1853.

CENTER, N. R. 2014. Introduction to Nondestructive Testing [Online]. Available: https://www.nde-ed.org/index_flash.htm [Accessed 2001-2014].

CHANG, S. I. & YADAMA, S. 2010. Statistical process control for monitoring non-linear profiles using wavelet filtering and B-spline approximation. International Journal of Production Research, 48, 1049-1068.

CHEN, C., KOVACEVIC, R. & JANDGRIC, D. 2003. Wavelet transform analysis of acoustic emission in monitoring friction stir welding of 6061 aluminum. International Journal of Machine Tools and Manufacture, 43, 1383-1390.

CHEN, G., MA, Q., ZHANG, S., WU, J., ZHANG, G. & SHI, Q. 2018. Computational fluid dynamics simulation of friction stir welding: A comparative study on different frictional boundary conditions. Journal of Materials Science & Technology, 34, 128-134.

CHEN, H. 2008. Diversity and regularization in neural network ensembles. University of Birmingham.

CHIN, H., DANAI, K. & HAYASHI, S. 1994. On-line tool breakage detection in turning: A multi-sensor method. Journal of engineering for industry, 116, 117.

CHOI, D., KWON, W. & CHU, C. 1999. Real-time monitoring of tool fracture in turning using sensor fusion. The International Journal of Advanced Manufacturing Technology, 15, 305-310.

CHOW, J. & WRIGHT, P. 1988. On-line estimation of tool/chip interface temperatures for a turning operation. Journal of Engineering for Industry, 110, 56-64.

COSANDEY-GODIN, A., KRAINSKI, E. T., WORM, B. & FLEMMING, J. M. 2014. Applying Bayesian spatiotemporal models to fisheries bycatch in the Canadian Arctic. Canadian Journal of Fisheries and Aquatic Sciences, 72, 186-197.

CRESSIE, N., SHI, T. & KANG, E. L. 2010. Fixed rank filtering for spatio-temporal data. Journal of Computational and Graphical Statistics, 19, 724-745.

DEMIRIZ, A., BENNETT, K. P. & SHAWE-TAYLOR, J. 2002. Linear programming boosting via column generation. Machine Learning, 46, 225-254.

DEVILLEZ, A., LESKO, S. & MOZER, W. 2004. Cutting tool crater wear measurement with white light interferometry. Wear, 256, 56-65.

DIEI, E. & DORNFELD, D. 1987. Acoustic emission sensing of tool wear in face milling. Journal of engineering for Industry, 109, 234-240.

DIETTERICH, T. G. 2000. Ensemble methods in machine learning. Multiple classifier systems, 1857, 1-15.

DIETTERICH, T. G. & BAKIRI, G. 1994. Solving multiclass learning problems via error-correcting output codes. Journal of artificial intelligence research, 2, 263-286.

DIMLA SNR, D. Tool-wear monitoring using cutting force measurements. ADVANCES IN MANUFACTURING TECHNOLOGY-CONFERENCE-, 1999. TAYLOR & FRANCIS LTD, 33-38.

DIMLA SR, D. & LISTER, P. 2000. On-line metal cutting tool condition monitoring.: I: force and vibration analyses. International Journal of Machine Tools and Manufacture, 40, 739-768.

DONOHO, D. L. & JOHNSTONE, J. M. 1994. Ideal spatial adaptation by wavelet shrinkage. biometrika, 81, 425-455.

DÖRFLER, S. M. Advanced modeling of friction stir welding–improved material model for aluminum alloys and modeling of different materials with different properties by using the level set method. COMSOL Conference Hannover, 2008.

EL-WARDANY, T., GAO, D. & ELBESTAWI, M. 1996. Tool condition monitoring in drilling using vibration signature analysis. International Journal of Machine Tools and Manufacture, 36, 687-711.

ELANGOVAN, K., BALASUBRAMANIAN, V. & VALLIAPPAN, M. 2008. Effect of tool pin profile and tool rotational speed on mechanical properties of friction stir welded AA6061 aluminium alloy. Materials and Manufacturing Processes, 23, 251-260.

FAN, J. & GIJBELS, I. 1996. Local polynomial regression and its applications. London: Chapman and Hall.

FAN, J., ZHANG, C. & ZHANG, J. 2001. Generalized likelihood ratio statistics and Wilks phenomenon. Annals of statistics, 153-193.

FEHRENBACHER, A., DUFFIE, N. A., FERRIER, N. J., PFEFFERKORN, F. E. & ZINN, M. R. 2011. Toward automation of friction stir welding through temperature measurement and closed-loop control. Journal of Manufacturing Science and Engineering, 133, 051008.

FREUND, Y. 2001. An adaptive version of the boost by majority algorithm. Machine learning, 43, 293-318.

FREUND, Y. 2009. A more robust boosting algorithm. arXiv preprint arXiv:0905.2138.

FRIEDMAN, J., HASTIE, T. & TIBSHIRANI, R. 2000. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). The annals of statistics, 28, 337-407.

FRIEDMAN, J., HASTIE, T. & TIBSHIRANI, R. 2001. The elements of statistical learning, Springer series in statistics New York.

FRIEDMAN, J. H. 1989. Regularized discriminant analysis. Journal of the American statistical association, 84, 165-175.

FU, P., HOPE, A. & KING, G. 1998a. Intelligent tool condition monitoring in milling operation. SOUTHAMPTON INST (UNITED KINGDOM) SYSTEMS ENGINEERING FACULTY.

FU, P., HOPE, A. D. & KING, G. An intelligent pattern recognition algorithm and its application in cutting tool condition monitoring process. International Conference on Information Technology for Balanced Automation Systems, 1998b. Springer, 515-524.
GAO, D., ERSOY, U., STEVENSON, R. & WANG, P.-C. 2009. A new one-sided joining process for aluminum alloys: friction stir blind riveting. Journal of Manufacturing Science and Engineering, 131, 061002.

GAO, Y. & DURRANT-WHYTE, H. F. Multi-sensor fault detection and diagnosis using combined qualitative and quantitative techniques. Multisensor Fusion and Integration for Intelligent Systems, 1994. IEEE International Conference on MFI'94., 1994. IEEE, 43-50.

GAO, Z., GUO, W. & LI, J. Sensor Fusion and On-Line Monitoring of Friction Stir Blind Riveting for Lightweight Materials Manufacturing. ASME 2018 13th International Manufacturing Science and Engineering Conference, 2018. American Society of Mechanical Engineers, V003T02A011-V003T02A011.

GARDNER, M. M., LU, J.-C., GYURCSIK, R. S., WORTMAN, J. J., HORNUNG, B. E., HEINISCH, H. H., RYING, E. A., RAO, S., DAVIS, J. C. & MOZUMDER, P. K. 1997. Equipment fault detection using spatial signatures. IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part C, 20, 295-304.

GEMAN, S., BIENENSTOCK, E. & DOURSAT, R. 1992. Neural networks and the bias/variance dilemma. Neural computation, 4, 1-58.

GIBSON, B., LAMMLEIN, D., PRATER, T., LONGHURST, W., COX, C., BALLUN, M., DHARMARAJ, K., COOK, G. & STRAUSS, A. 2014a. Friction stir welding: process, automation, and control. Journal of Manufacturing Processes, 16, 56-73.

GIBSON, B. T., LAMMLEIN, D., PRATER, T., LONGHURST, W., COX, C., BALLUN, M., DHARMARAJ, K., COOK, G. & STRAUSS, A. 2014b. Friction stir welding: Process, automation, and control. Journal of Manufacturing Processes, 16, 56-73.

GORDON, N. J., SALMOND, D. J. & SMITH, A. F. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE proceedings F (radar and signal processing), 1993. IET, 107-113.

GRASSO, M., COLOSIMO, B. & PACELLA, M. 2014. Profile monitoring via sensor fusion: the use of PCA methods for multi-channel data. International Journal of Production Research, 52, 6110-6135.

GREGORY, M. A. & ROEBROEKS, G. H. J. J. 1991. Fiber-metal laminates: A solution to weight, strength, and fatigue problems. In National Research Council of Canada, Airframe Materials p 43-56(SEE N 93-23935 08-24), 1991.

GUO, W., JIN, J. J. & HU, S. J. Profile Monitoring and Fault Diagnosis via Sensor Fusion for Ultrasonic Welding. ASME 2016 11th International Manufacturing Science and Engineering Conference, 2016. American Society of Mechanical Engineers, V002T04A028-V002T04A028.

GUO, W. G., CHEN, J., GUO, S. & LI, J. Process monitoring of friction stir blind riveting for lightweight materials. 67th Annual Conference and Expo of the Institute of Industrial Engineers 2017, 2017. Institute of Industrial Engineers.

GUPTA, S., MONTGOMERY, D. & WOODALL, W. 2006. Performance evaluation of two methods for online monitoring of linear calibration profiles. International journal of production research, 44, 1927-1942.

HAARIO, H., LAINE, M., MIRA, A. & SAKSMAN, E. 2006. DRAM: efficient adaptive MCMC. Statistics and computing, 16, 339-354.

HABER, R. E., JIMÉNEZ, J. E., PERES, C. R. & ALIQUE, J. R. 2004. An investigation of tool-wear monitoring in a high-speed machining process. Sensors and Actuators A: Physical, 116, 539-545.

HANSEN, L. K. & SALAMON, P. 1990. Neural network ensembles. IEEE transactions on pattern analysis and machine intelligence, 12, 993-1001.

HIRASAWA, S., BADARINARAYAN, H., OKAMOTO, K., TOMIMURA, T. & KAWANAMI, T. 2010. Analysis of effect of tool geometry on plastic flow during friction stir spot welding using particle method. Journal of materials processing technology, 210, 1455-1463.

HO, T. K. 1998. The random subspace method for constructing decision forests. IEEE transactions on pattern analysis and machine intelligence, 20, 832-844.

HOLTON, J. R. 1973. An introduction to dynamic meteorology. American Journal of Physics, 41, 752-754.

IHN, J.-B. & CHANG, F.-K. 2004. Detection and monitoring of hidden fatigue crack growth using a built-in piezoelectric sensor/actuator network: II. Validation using riveted joints and repair patches. Smart materials and structures, 13, 621.

IMMARIGEON, J. P., HOLT, R. T., KOUL, A. K., ZHAO, L., WALLACE, W. & BEDDOES, J. C. 1995. Lightweight materials for aircraft applications. Materials Characterization, 35, 41-67.

JACOBS, R. A., JORDAN, M. I., NOWLAN, S. J. & HINTON, G. E. 1991. Adaptive mixtures of local experts. Neural computation, 3, 79-87.

JANTUNEN, E. 2002. A summary of methods applied to tool condition monitoring in drilling. International Journal of Machine Tools and Manufacture, 42, 997-1010.

JEMIELNIAK, K. & OTMAN, O. 1998. Tool failure detection based on analysis of acoustic emission signals. Journal of Materials Processing Technology, 76, 192-197.

JENSEN, W. A., BIRCH, J. B. & WOODALL, W. H. 2008. Monitoring correlation within linear profiles using mixed models. Journal of Quality Technology, 40, 167-183.
JEONG, M. K., LU, J.-C., HUO, X., VIDAKOVIC, B. & CHEN, D. 2006a. Wavelet-based data reduction techniques for process fault detection. Technometrics, 48, 26-40.

JEONG, M. K., LU, J.-C. & WANG, N. 2006b. Wavelet-based SPC procedure for complicated functional data. International Journal of Production Research, 44, 729-744.

JEONG, Y.-S., JEONG, M. K., LU, J.-C., YUAN, M. & JIN, J. 2018. Statistical process control procedures for functional data with systematic local variations. IISE Transactions, 50, 448-462.

JIANG, S.-B., YANG, C.-M., HUANG, R.-S., FANG, C.-Y. & YEH, T.-L. 2011. An innovative ultrasonic time-of-flight measurement method using peak time sequences of different frequencies: Part I. IEEE Transactions on Instrumentation and Measurement, 60, 735-744.

JIN, J. & SHI, J. 2001. Automatic feature extraction of waveform signals for in-process diagnostic performance improvement. Journal of Intelligent Manufacturing, 12, 257-268.

JUNG, U., JEONG, M. K. & LU, J.-C. 2006a. Data reduction for multiple functional data with class information. International journal of production research, 44, 2695-2710.

JUNG, U., JEONG, M. K. & LU, J.-C. 2006b. A vertical-energy-thresholding procedure for data reduction with multiple complex curves. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 36, 1128-1138.

KAHL, S. 2010. The influence of small voids on the fatigue strength of friction stir welds in the aluminium alloy AA6061-T6. Heron, 55, 4.

KANG, L. & ALBIN, S. L. 2000. On-line monitoring when the process yields a linear profile. Journal of quality Technology, 32, 418.

KANNATEY-ASIBU, E. & DORNFELD, D. A. 1981. Quantitative relationships for acoustic emission from orthogonal metal cutting. Journal of Engineering for Industry, 103, 330-340.

KATZFUSS, M. & CRESSIE, N. 2012. Bayesian hierarchical spatio-temporal smoothing for very large datasets. Environmetrics, 23, 94-107.

KIM, J., HUANG, Q., SHI, J. & CHANG, T.-S. 2006. Online multichannel forging tonnage monitoring and fault pattern discrimination using principal curve. Journal of manufacturing science and engineering, 128, 944-950.

KIM, K., MAHMOUD, M. A. & WOODALL, W. H. 2003. On the monitoring of linear profiles. Journal of Quality Technology, 35, 317.

KINCHEN, D. G. & ALDAHIR, E. 2002. NDE of friction stir welds in aerospace applications.

KOLDA, T. G. & BADER, B. W. 2009. Tensor decompositions and applications. SIAM review, 51, 455-500.

KONRAD, H., ISERMANN, R. & OETTE, H. 1994. Supervision of tool wear and surface quality during end milling operations. IFAC Proceedings Volumes, 27, 507-512.

KROGH, A. & VEDELSBY, J. Neural network ensembles, cross validation, and active learning. Advances in neural information processing systems, 1995. 231-238.

KUMAR, A. & RAJU, L. S. 2012. Influence of tool pin profiles on friction stir welding of copper. Materials and Manufacturing Processes, 27, 1414-1418.

KURADA, S. & BRADLEY, C. 1997. A review of machine vision sensors for tool condition monitoring. Computers in industry, 34, 55-72.

KYRIAKIDIS, P. C. & JOURNEL, A. G. 1999. Geostatistical space–time models: a review. Mathematical geology, 31, 651-684.

LAMARRE, A., MOLES, M. & LUPIEN, V. Phased array ultrasonic inspection of friction stir weldments. AIP Conference Proceedings, 2000. AIP, 1333-1340.

LATHABAI, S., TYAGI, V., RITCHIE, D., KEARNEY, T., FINNIN, B., CHRISTIAN, S., SANSOME, A. & WHITE, G. 2011. Friction stir blind riveting: a novel joining process for automotive light alloys. SAE International Journal of Materials and Manufacturing, 4, 589-601.

LEE, Y.-L. 2005. Fatigue testing and analysis: theory and practice, Butterworth-Heinemann.

LEI, Y., ZHANG, Z. & JIN, J. 2010. Automatic tonnage monitoring for missing part detection in multi-operation forging processes. Journal of Manufacturing Science and Engineering, 132, 051010.

LI, B., SHEN, Y. & HU, W. 2011. The study on defects in aluminum 2219-T6 thick butt friction stir welds with the application of multiple non-destructive testing methods. Materials & Design, 32, 2073-2084.

LI, C. J. & LI, S. 1993. A new sensor for real-time milling tool condition monitoring. Journal of dynamic systems, measurement, and control, 115, 285-290.

LI, F., ROGERS, W., BUSTA, J. A., CHILKO, M. R. & SCHAPER, G. A. 2015. Evaluation of porosity in aluminum ingot. Google Patents.

LI, S., KHAN, H., HIHARA, L. H. & LI, J. 2016. Marine atmospheric corrosion of Al-Mg joints by friction stir blind riveting. Corrosion Science, 111, 793-801.

LINDGREN, F., RUE, H. & LINDSTRÖM, J. 2011. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 73, 423-498.

LITTLESTONE, N. & WARMUTH, M. K. 1994. The weighted majority algorithm. Information and computation, 108, 212-261.

LIU, J. S. 2008. Monte Carlo strategies in scientific computing, Springer Science & Business Media.

LIU, Y. & YAO, X. 1999. Ensemble learning via negative correlation. Neural networks, 12, 1399-1404.

LU, H. 2008. Multilinear subspace learning for face and gait recognition, Citeseer.

LU, H., PLATANIOTIS, K. N. & VENETSANOPOULOS, A. 2013. Multilinear subspace learning: dimensionality reduction of multidimensional data, CRC press.

LU, H., PLATANIOTIS, K. N. & VENETSANOPOULOS, A. N. 2008a. MPCA: Multilinear principal component analysis of tensor objects. IEEE Transactions on Neural Networks, 19, 18-39.

LU, H., PLATANIOTIS, K. N. & VENETSANOPOULOS, A. N. Uncorrelated multilinear principal component analysis through successive variance maximization. Proceedings of the 25th international conference on Machine learning, 2008b. ACM, 616-623.

LU, H., PLATANIOTIS, K. N. & VENETSANOPOULOS, A. N. 2009. Uncorrelated multilinear discriminant analysis with regularization and aggregation for tensor object recognition. IEEE Transactions on Neural Networks, 20, 103-123.

LUCAS, J. M. & SACCUCCI, M. S. 1990. Exponentially weighted moving average control schemes: properties and enhancements. Technometrics, 32, 1-12.

MAHALANOBIS, P. C. On the generalized distance in statistics. 1936. National Institute of Science of India.

MAHMOUD, M. A., PARKER, P. A., WOODALL, W. H. & HAWKINS, D. M. 2007. A change point method for linear profile data. Quality and Reliability Engineering International, 23, 247-268.

MAHMOUD, M. A. & WOODALL, W. H. 2004. Phase I analysis of linear profiles with calibration applications. Technometrics, 46, 380-391.

MARTIN, K., BRANDON, J., GROSVENOR, R. & OWEN, A. A comparison of in-process tool wear measurement methods in turning. Proceedings of the Twenty-Sixth International Machine Tool Design and Research Conference, 1986. Springer, 289-296.

MAURER, D., ROESER, H., OPPER, R., WOJCIK, A. & SCHOENIG, C. 2002. Riveting system and process for forming a riveted joint. Google Patents.
MIN, J., LI, J., CARLSON, B. E., LI, Y. & LIN, J. 2014. Mechanical property of Al alloy joints by friction stir blind riveting. Procedia engineering, 81, 2036-2041.

MIN, J., LI, J., CARLSON, B. E., LI, Y., QUINN, J. F., LIN, J. & WANG, W. 2015a. Friction Stir Blind Riveting for Joining Dissimilar Cast Mg AM60 and Al Alloy Sheets. Journal of Manufacturing Science and Engineering, 137, 051022-051022-9.

MIN, J., LI, J., LI, Y., CARLSON, B. E., LIN, J. & WANG, W.-M. 2015b. Friction stir blind riveting for aluminum alloy sheets. Journal of Materials Processing Technology, 215, 20-29.

MIN, J., LI, Y., CARLSON, B. E., HU, S. J., LI, J. & LIN, J. 2015c. A new single-sided blind riveting method for joining dissimilar materials. CIRP Annals, 64, 13-16.

MIN, J., LI, Y., LI, J., CARLSON, B. E. & LIN, J. 2015d. Friction stir blind riveting of carbon fiber-reinforced polymer composite and aluminum alloy sheets. The International Journal of Advanced Manufacturing Technology, 76, 1403-1410.

MOSESOVA, S. A., CHIPMAN, H. A., MACKAY, R. J. & STEINER, S. H. 2006. Profile monitoring using mixed-effects models. Technical Report 06–06.

MYERS, M., WALKER, D., YUHAS, D. & MUTTON, M. Heat flux determination from ultrasonic pulse measurements. ASME 2008 International Mechanical Engineering Congress and Exposition, 2008. American Society of Mechanical Engineers, 739-743.

NOOROSSANA, R., SAGHAEI, A. & AMIRI, A. 2011. Statistical analysis of profile monitoring, John Wiley & Sons.

PAI, P. S. & RAO, P. R. 2002. Acoustic emission analysis for tool wear monitoring in face milling. International Journal of Production Research, 40, 1081-1093.

PAYNABAR, K. & JIN, J. 2011. Characterization of non-linear profiles variations using mixed-effect models and wavelets. IIE Transactions, 43, 275-290.

PAYNABAR, K., JIN, J., AGAPIOU, J. & DEEDS, P. 2012. Robust leak tests for transmission systems using nonlinear mixed-effect models. Journal of Quality Technology, 44, 265-278.

PAYNABAR, K., JIN, J. & PACELLA, M. 2013. Monitoring and diagnosis of multichannel nonlinear profile variations using uncorrelated multilinear principal component analysis. IIE Transactions, 45, 1235-1247.

PETRIS, G., PETRONE, S. & CAMPAGNOLI, P. 2009. Dynamic linear models. Dynamic Linear Models with R. Springer.

PFEIFER, T. & WIEGERS, L. 2000. Reliable tool wear monitoring by optimized image and illumination control in machine vision. Measurement, 28, 209-218.

PRADO, R., MURR, L., SHINDO, D. & SOTO, K. 2001. Tool wear in the friction-stir welding of aluminum alloy 6061+ 20 al 2 o 3: A preliminary study. Scripta materialia, 1, 75-80.

PRADO, R., MURR, L., SOTO, K. & MCCLURE, J. 2003. Self-optimization in tool wear for friction-stir welding of Al 6061+ 20% Al2O3 MMC. Materials Science and Engineering: A, 349, 156-165.

QIU, P., ZOU, C. & WANG, Z. 2010. Nonparametric profile monitoring by mixed effects modeling. Technometrics, 52, 265-277.

QUAN, R., QUAN, S., HUANG, L. & XIE, C. 2010. Fault diagnosis and fault-tolerant control for multi-sensor of fuel cell system using two-level neural networks. Journal of Computational Information Systems, 6, 3703-3711.

RAI, R., DE, A., BHADESHIA, H. & DEBROY, T. 2011. friction stir welding tools. Science and Technology of welding and Joining, 16, 325-342.

RUE, H. & HELD, L. 2005. Gaussian Markov random fields: theory and applications, Chapman and Hall/CRC.

RUE, H., MARTINO, S. & CHOPIN, N. 2009. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. Journal of the royal statistical society: Series b (statistical methodology), 71, 319-392.

RUTA, D. & GABRYS, B. 2000. An overview of classifier fusion methods. Computing and Information systems, 7, 1-10.

SAINI, D. & FLOYD, S. 1998. An investigation of gas metal arc welding sound signature for on-line quality control. WELDING JOURNAL-NEW YORK-, 77, 172-s.

SALAHSHOOR, K., MOSALLAEI, M. & BAYAT, M. 2008. Centralized and decentralized process and sensor fault monitoring using data fusion based on adaptive extended Kalman filter algorithm. Measurement, 41, 1059-1076.

SANTOS, M. & SANTOS, J. 2010. Lamb waves technique applied to the characterization of defects in friction stir welding of aluminum plates: comparison with X-ray and ultrasonic C-scan. Journal of Testing and Evaluation, 38, 622-627.

SATO, Y. S., URATA, M. & KOKAWA, H. 2002. Parameters controlling microstructure and hardness during friction-stir welding of precipitation-hardenable aluminum alloy 6063. Metallurgical and Materials Transactions A, 33, 625-635.

SCHAPIRE, R. E. 1990. The strength of weak learnability. Machine learning, 5, 197-227.

SCHAPIRE, R. E. 2003. The boosting approach to machine learning: An overview. Nonlinear estimation and classification. Springer.
SCHAPIRE, R. E., FREUND, Y., BARTLETT, P. & LEE, W. S. 1998. Boosting the margin: A new explanation for the effectiveness of voting methods. The annals of statistics, 26, 1651-1686.

SCHAPIRE, R. E. & SINGER, Y. 1999. Improved boosting algorithms using confidence-rated predictions. Machine learning, 37, 297-336.

SCIALPI, A., DE GIORGI, M., DE FILIPPIS, L., NOBILE, R. & PANELLA, F. 2008. Mechanical analysis of ultra-thin friction stir welding joined sheets with dissimilar and similar materials. Materials & Design, 29, 928-936.

SHAO, C., GUO, W., KIM, T. H., JIN, J. J., HU, S. J., SPICER, J. P. & ABELL, J. A. Characterization and monitoring of tool wear in ultrasonic metal welding. 9th International Workshop on Microfactories, 2014. 161-169.

SHAO, C., KIM, T. H., HU, S. J., JIN, J. J., ABELL, J. A. & SPICER, J. P. 2016. Tool wear monitoring for ultrasonic metal welding of lithium-ion batteries. Journal of Manufacturing Science and Engineering, 138, 051005.

SHINDO, D., RIVERA, A. & MURR, L. 2002. Shape optimization for tool wear in the friction-stir welding of cast AI359-20% SiC MMC. Journal of materials science, 37, 4999-5005.

SIMON, N., FRIEDMAN, J., HASTIE, T. & TIBSHIRANI, R. 2013. A sparse-group lasso. Journal of Computational and Graphical Statistics, 22, 231-245.

SITHOLE, K. & RAO, V. V. Recent developments in micro friction stir welding: a review. IOP conference series: materials science and engineering, 2016. IOP Publishing, 012036.

SMITH, A. 2013. Sequential Monte Carlo methods in practice, Springer Science & Business Media.

STROUD, J. R., STEIN, M. L., LESHT, B. M., SCHWAB, D. J. & BELETSKY, D. 2010. An ensemble Kalman filter and smoother for satellite data assimilation. Journal of the American Statistical Association, 105, 978-990.

SU, H., WU, C. S., BACHMANN, M. & RETHMEIER, M. 2015. Numerical modeling for the effect of pin profiles on thermal and material flow characteristics in friction stir welding. Materials & Design, 77, 114-125.

SUITS, M. W., LEAK, J. & BRYSON, C. 2003. Nondestructive Inspection Techniques for Friction Stir Weld verification on the space shuttle external tank.

TAO, D., LI, X., WU, X. & MAYBANK, S. J. 2007. General tensor discriminant analysis and gabor features for gait recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29.

THOMAS, W., NICHOLAS, E., NEEDHAM, J., MURCH, M., TEMPLESMITH, P. & DAWES, C. 1991. International patent application no. PCT/GB92/02203 and GB patent application.

THOMPSON, B. T. 2010. Tool degradation characterization in the friction stir welding of hard metals. The Ohio State University.

THREADGILL, P., LEONARD, A., SHERCLIFF, H. & WITHERS, P. 2009. Friction stir welding of aluminium alloys. International Materials Reviews, 54, 49-93.

TIKHONOV, A. N., GONCHARSKY, A., STEPANOV, V. & YAGOLA, A. G. 2013. Numerical methods for the solution of ill-posed problems, Springer Science & Business Media.

TWI 2019. DEVELOPMENTS IN MICRO APPLICATIONS OF FRICTION STIR WELDING.

VAGHEFI, A., TAJBAKHSH, S. D. & NOOROSSANA, R. 2009. Phase II monitoring of nonlinear profiles. Communications in Statistics—Theory and Methods, 38, 1834-1851.

VILLAGRAN, A., HUERTA, G., JACKSON, C. S. & SEN, M. K. 2008. Computational methods for parameter estimation in climate models. Bayesian Analysis, 3, 823-850.

VRUGT, J. A., TER BRAAK, C., DIKS, C., ROBINSON, B. A., HYMAN, J. M. & HIGDON, D. 2009. Accelerating Markov chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling. International Journal of Nonlinear Sciences and Numerical Simulation, 10, 273-290.

WALKER, E. & WRIGHT, S. P. 2002. Comparing curves using additive models. Journal of Quality Technology, 34, 118.

WANG, K., KHAN, H. A., LI, Z., LYU, S. & LI, J. 2018. Micro friction stir welding of multilayer aluminum alloy sheets. Journal of Materials Processing Technology, 260, 137-145.

WANG, K., REIF, G. & OH, S. 1982. In-process quality detection of friction welds using acoustic emission techniques. WELDING J., 61, 312.

WANG, W.-M., ALI KHAN, H., LI, J., MILLER, S. F. & ZACHARY TRIMBLE, A. 2016. Classification of Failure Modes in Friction Stir Blind Riveted Lap-Shear Joints With Dissimilar Materials. Journal of Manufacturing Science and Engineering, 139, 021005-021005-10.

WANG, Y. & GONG, S. Tensor discriminant analysis for view-based object recognition. Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, 2006. IEEE, 33-36.

WARMUTH, M. K., LIAO, J. & RÄTSCH, G. Totally corrective boosting algorithms that maximize the margin. Proceedings of the 23rd international conference on Machine learning, 2006. ACM, 1001-1008.

WEINBERGER, T., KHOSA, S. U., FÜHRER, B. & ENZINGER, N. Analysis of tool wear and failure mechanism during friction stir welding of steel. 7th International Symposium Friction Stir Welding, 2008. .

WHITTLE, P. 1954. On stationary processes in the plane. Biometrika, 434-449.

WIKLE, C. K., BERLINER, L. M. & CRESSIE, N. 1998. Hierarchical Bayesian space-time models. Environmental and Ecological Statistics, 5, 117-154.

WIKLE, C. K. & HOOTEN, M. B. 2010. A general science-based framework for dynamical spatio-temporal models. Test, 19, 417-451.

WIKLE, C. K., MILLIFF, R. F., NYCHKA, D. & BERLINER, L. M. 2001. Spatiotemporal hierarchical Bayesian modeling tropical ocean surface winds. Journal of the American Statistical Association, 96, 382-397.

WREDENBERG, F. & LARSSON, P.-L. 2009. Scratch testing of metals and polymers: experiments and numerics. Wear, 266, 76-83.

WU, H. & ZHANG, J.-T. 2002. Local polynomial mixed-effects models for longitudinal data. Journal of the American Statistical Association, 97, 883-897.

WU, P. & STEPINSKI, T. 1998. Inspection of copper canisters for spent nuclear fuel by means of Ultrasonic array System. Modelling, defect detection and grain noise estimation. Swedish Nuclear Fuel and Waste Management Co.

XU, L., JORDAN, M. I. & HINTON, G. E. An alternative model for mixtures of experts. Advances in neural information processing systems, 1995. 633-640.

YAN, H., PAYNABAR, K. & SHI, J. 2015. Image-based process monitoring using low-rank tensor decomposition. IEEE Transactions on Automation Science and Engineering, 12, 216-227.

YAN, S., XU, D., YANG, Q., ZHANG, L., TANG, X. & ZHANG, H.-J. 2007. Multilinear discriminant analysis for face recognition. IEEE Transactions on Image Processing, 16, 212-220.

YANG, J., ZHANG, D., FRANGI, A. F. & YANG, J.-Y. 2004. Two-dimensional PCA: a new approach to appearance-based face representation and recognition. IEEE transactions on pattern analysis and machine intelligence, 26, 131-137.

YE, J. 2005. Generalized low rank approximations of matrices. Machine Learning, 61, 167-191.

YE, J., JANARDAN, R. & LI, Q. GPCA: an efficient dimension reduction scheme for image compression and retrieval. Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 2004. ACM, 354-363.

YE, J., JANARDAN, R. & LI, Q. Two-dimensional linear discriminant analysis. Advances in neural information processing systems, 2005. 1569-1576.

YUAN, M. & LIN, Y. 2006. Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 68, 49-67.

ZHANG, C. Q., WANG, X. J. & LI, B. Q. A technological study on friction stir blind rivet jointing of AZ31B magnesium alloys and high-strength DP600 steel. Advanced Materials Research, 2011. Trans Tech Publ, 1616-1620.

ZHANG, Y., ZHANG, K. & HAN, Z. Detection of tool breakage in turning operations by using neural networks. International Conference on Intelligent Manufacturing, 1995. International Society for Optics and Photonics, 463-468.

ZHOU, Z.-H. 2012. Ensemble methods: foundations and algorithms, CRC press.

ZOU, C., QIU, P. & HAWKINS, D. 2009. Nonparametric control chart for monitoring profiles using change point formulation and adaptive smoothing. Statistica Sinica, 1337-1357.

ZOU, C., TSUNG, F. & WANG, Z. 2008. Monitoring profiles based on nonparametric regression methods. Technometrics, 50, 512-526.

ZOU, C., ZHANG, Y. & WANG, Z. 2006. A control chart based on a change-point model for monitoring linear profiles. IIE transactions, 38, 1093-1103.